

3/3/5 (Item 5 from file: 351)
DIALOG(R) File 351:Derwent WPI
(c) 2006 Thomson Derwent. All rts. reserv.

013536244 **Image available**
WPI Acc No: 2001-020450/ 200103
XRPX Acc No: N01-015720

Juxtaposition calculation apparatus for physical property estimation, has
data converters which do not change symbol and exponent parts of data
before and after conversion, and change only mantissa part

Patent Assignee: FUJI XEROX CO LTD (XERF); TAISHO PHARM CO LTD (TAIS)
Inventor: AMISAKI T; INABATA S; KITAMURA K; MIYAKAWA N; TAKASHIMA H; YAMADA
S

Number of Countries: 002 Number of Patents: 002

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
JP 2000293494	A	20001020	JP 99102020	A	19990409	200103 B
US 6631391	B1	20031007	US 2000544359	A	20000406	200374

Priority Applications (No Type Date): JP 99102020 A 19990409

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
JP 2000293494	A		25	G06F-015/16	
US 6631391	B1			G06F-007/38	

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-293494

(43)Date of publication of application : 20.10.2000

(51)Int.Cl.

G06F 15/16

G06F 7/00

G06F 9/38

(21)Application number : 11-102020

(71)Applicant : FUJI XEROX CO LTD
TAISHO PHARMACEUT CO LTD

(22)Date of filing : 09.04.1999

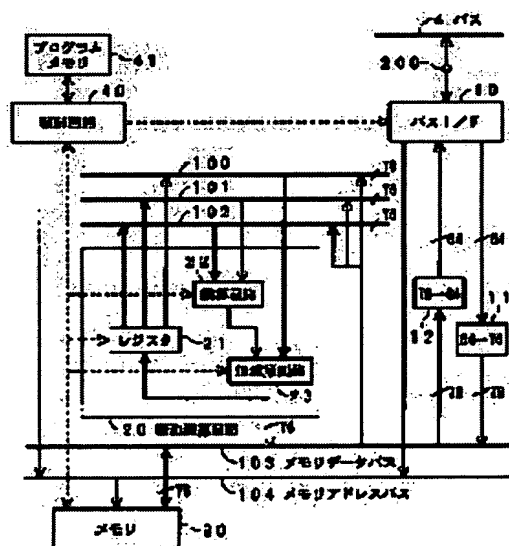
(72)Inventor : INAHATA SHINJIROU
YAMADA SOU
MIYAGAWA NOBUAKI
AMISAKI TAKASHI
TAKASHIMA HAJIME
KITAMURA KAZUYASU

(54) DEVICE AND METHOD FOR PARALLEL COMPUTATION

(57)Abstract:

PROBLEM TO BE SOLVED: To eliminate the need for an overspecification hardware scale, to maintain a high calculation speed, and to enable high-precision parallel operations by converting floating-point data in 2nd format into floating-point data in 1st format.

SOLUTION: A host processor performs arithmetic processing as to the 1st format, i.e., double-precision floating-point data and a processor put in partial charge of floating-point arithmetic processing with the host processor performs the floating-point operation by using floating-point data in 2nd format which has higher precision than the floating-point data in 1st format and is different in only the bit width of a mantissa part. Each time input data is received from the host processor, the processor element converts the format of the floating-point data from the 1st format to the 2nd format by an input data conversion part 11. Further, when data is outputted from the processor element to the host, an output data conversion part 12 converts the format of the floating-point data from the 2nd format to the 1st format.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision
of rejection]

[Date of requesting appeal against examiner's
decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号
特開2000-293494
(P2000-293494A)

(43) 公開日 平成12年10月20日 (2000. 10. 20)

(51) Int.Cl. ⁷	識別記号	F I	テマコード [*] (参考)
G 0 6 F 15/16	6 1 0	G 0 6 F 15/16	6 1 0 Z 5 B 0 1 3
7/00		9/38	3 7 0 C 5 B 0 2 2
9/38	3 7 0	7/00	1 0 1 Z 5 B 0 4 5

審査請求 未請求 請求項の数 9 O L (全 25 頁)

(21) 出願番号 特願平11-102020

(22) 出願日 平成11年4月9日 (1999. 4. 9)

(71) 出願人 000005496

富士ゼロックス株式会社
東京都港区赤坂二丁目17番22号

(71) 出願人 000002819

大正製薬株式会社
東京都豊島区高田3丁目24番1号

(72) 発明者 稲畑 深二郎

神奈川県足柄上郡中井町境430 グリーン
テクなかい富士ゼロックス株式会社内

(74) 代理人 100091546

弁理士 佐藤 正美

最終頁に続く

(54) 【発明の名称】 並列計算装置および並列計算方法

(57) 【要約】

【課題】 オーバースペックになるようなハードウェア規模とする必要がなく、また、計算速度も高速を保ちながら、高精度の並列計算ができる。

【解決手段】 ホストプロセッサと、複数個の専用プロセッサとがバスを介して接続されてシステムが構成される。ホストプロセッサでは、倍精度浮動小数点のフォーマットで演算を行い、専用プロセッサでは、それより高精度の浮動小数点の内部フォーマットで演算を行う。内部フォーマットと、倍精度浮動小数点データとは、符号部と、指数部とが共通の構成を有し、仮数部のビット幅が内部フォーマットの方が大きく設定する。専用プロセッサでは、倍精度から内部フォーマットへの変換を行う入力データ変換部と、内部フォーマットから倍精度への変換を行う出力データ変換部を備える。これらデータ変換部では、変換の前後のデータにおいて、符号部および指数部は、共通にデータを使用することができるので、仮数部のみを、特定の手順で変換を行う。

(A) IEEE規格754の倍精度表現

(ビット数) 1 11 52

s	e	f 0
---	---	-----

$$v = (-1)^s \times 2^{e-1023} \times (1 + f0 \times 2^{-52}) \quad (0 < e < 2047)$$

$$= \text{NaN (不定)} \quad (e = 2047, f0 \neq 0)$$

$$= (-1)^s \times \infty \quad (e = 2047, f0 = 0)$$

$$= (-1)^s \times 2^{e-1022} (f0 \times 2^{-52}) \quad (e = 0, f0 \neq 0)$$

$$= 0 \quad (e = 0, f0 = 0)$$

(B) 専用プロセッサ2内での76ビット内部フォーマット表現

(ビット数) 1 11 64

s	e	f 1
---	---	-----

$$v = (-1)^s \times 2^{e-1023} \times (1 + f1 \times 2^{-64}) \quad (0 < e < 2047)$$

$$= (-1)^s \times \infty \quad (e = 2047)$$

$$= 0 \quad (e = 0)$$

【特許請求の範囲】

【請求項 1】 ホストプロセッサと、このホストプロセッサとバスを介して接続される 1 または複数のプロセッサエレメントとからなり、前記ホストプロセッサと、前記 1 または複数のプロセッサエレメントとが、特定の計算プロセスに含まれる浮動小数点演算処理を分担して計算する並列計算装置において、

前記プロセッサエレメントは、

前記ホストプロセッサとの間で、1 ビットの符号部と、 m ビットのビット幅の指数部と、 $m0$ ビットのビット幅の仮数部とからなる第 1 の形式の浮動小数点データを入力する浮動小数点入出力インタフェース部と、

前記浮動小数点入出力インタフェース部より入力される前記第 1 の形式の浮動小数点データを、1 ビットの符号部と、前記 m ビットのビット幅の指数部と、前記 $m0$ よりも大きい $m1$ ビットのビット幅の仮数部とからなる第 2 の形式の浮動小数点データに変換する入力データ変換部と、

前記入力データ変換部からの前記第 2 の形式の浮動小数点データについて、浮動小数点演算処理を実行する浮動小数点演算部と、

前記浮動小数点演算部で演算が施された前記第 2 の形式の浮動小数点データを前記第 1 の形式の浮動小数点データに変換して、前記浮動小数点入出力インタフェース部に供給する出力データ変換部と、

を備えることを特徴とする並列計算装置。

【請求項 2】 請求項 1 に記載の並列計算装置において、

前記入力データ変換部は、

前記第 1 の形式の浮動小数点データの符号部を、前記第 2 の形式の浮動小数点データの符号部とし、

前記第 1 の形式の浮動小数点データの指数部を、前記第 2 の形式の浮動小数点データの指数部とし、

前記 $m0$ ビットの前記第 1 の形式の浮動小数点データの仮数部を、前記 $m1$ ビットの前記第 2 の形式の浮動小数点データの仮数部のうちの上位の前記 $m0$ ビットとし、前記第 2 の形式の浮動小数点データの仮数部の下位 $m1 - m0$ ビットを所定の数値とし、

前記出力データ変換部は、

前記第 2 の形式の浮動小数点データの符号部を、前記第 1 の形式の浮動小数点データの符号部とし、

前記第 2 の形式の浮動小数点データの指数部を、前記第 1 の形式の浮動小数点データの指数部とし、

前記第 2 の形式の浮動小数点データの仮数部の上位の前記 $m0$ ビットを、前記第 1 の形式の浮動小数点データの仮数部としたことを特徴とする並列計算装置。

【請求項 3】 請求項 1 に記載の並列計算装置において、

前記入力データ変換部は、

前記第 1 の形式の浮動小数点データの符号部を、前記第 2 の形式の浮動小数点データの符号部とし、

前記第 1 の形式の浮動小数点データの指数部を、前記第

2 の形式の浮動小数点データの指数部とし、

前記 $m1$ ビットの前記第 2 の形式の浮動小数点データの仮数部のうち、その最上位 1 ビットは 1 とし、最上位よりも下位の前記 $m0$ ビットは前記第 1 の形式の浮動小数点データの仮数部とし、さらにそれより下位の前記 $m1 - m0 - 1$ ビットは所定の数値とし、

前記出力データ変換部は、

前記第 2 の形式の浮動小数点データの符号部を、前記第 1 の形式の浮動小数点データの符号部とし、

10 前記第 2 の形式の浮動小数点データの指数部を、前記第 1 の形式の浮動小数点データの指数部とし、

前記第 2 の形式の浮動小数点データの仮数部の上位 2 ビット目からの $m0$ ビットを、前記 $m0$ ビットの第 1 の形式の浮動小数点データの仮数部としたことを特徴とする並列計算装置。

【請求項 4】 前記第 2 の形式の浮動小数点データにおいては、前記指数部が第 1 の所定の値 e_{max} のときに符号付き無限大を表し、前記指数部が第 2 の所定の値 e_{min} のときに零を表し、

20 前記浮動小数点演算部に含まれる浮動小数点乗算手段は、

オーバーフローが発生すると出力データの指数部を前記値 e_{max} とし、

アンダーフローが発生すると出力データの指数部を前記値 e_{min} とし、

入力される前記第 2 の形式の 2 つの浮動小数点データのうち、どちらか 1 つの指数部が前記値 e_{max} のときに出力データの指数部を前記値 e_{max} とし、

30 前記入力される前記第 2 の形式の 2 つの浮動小数点データのうち双方とも指数部が前記値 e_{max} でなく、かつ、

どちらか 1 つの指数部が前記値 e_{min} のときに出力データの指数部を前記値 e_{min} とし、

前記浮動小数点演算部に含まれる浮動小数点加減算手段は、

オーバーフローが発生すると出力データの指数部を前記値 e_{max} とし、

アンダーフローが発生すると出力データの指数部を前記値 e_{min} とし、

40 入力される前記第 2 の形式の 2 つの浮動小数点データのうち、どちらか 1 つの指数部が前記値 e_{max} のときに

出力データの指数部を前記値 e_{max} とすることを特徴とする請求項 1 ～ 3 のいずれかに記載の並列計算装置。

【請求項 5】 前記浮動小数点演算部に含まれる浮動小数点乗算手段は、その出力データの仮数部を求めるときに、切り捨てを実施し、

前記浮動小数点演算部に含まれる浮動小数点加減算手段は、その出力データの仮数部を求めるときに、切り捨てを実施することを特徴とする請求項 1 ～ 4 のいずれかに記載の並列計算装置。

50 【請求項 6】 ホストプロセッサと、このホストプロセッサとバスを介して接続される 1 または複数のプロセッサエレメントとからなり、前記ホストプロセッサと、前記 1 または複数のプロセッサエレメントとが、特定の計算プロセスに含まれる浮動小数点演算処理を分担して計算する並列計算装置において、

3

サとバスを介して接続される 1 または複数のプロセッサエレメントとにより、特定の計算プロセスに含まれる浮動小数点演算処理を分担して計算する並列計算方法において、

前記ホストプロセッサは、

1 ビットの符号部と、me ビットのビット幅の指数部と、m0 ビットのビット幅の仮数部とからなる第 1 の形式の浮動小数点データについて演算処理を行い、

前記プロセッサエレメントは、

前記第 1 の形式よりも高精度で、1 ビットの符号部と、前記me ビットのビット幅の指数部と、前記m0 よりも大きいm1 ビットのビット幅の仮数部とからなる第 2 の形式の浮動小数点データについて、浮動小数点演算処理を実行するものであって、かつ、

前記バスを通じて入力される前記第 1 の形式の浮動小数点データ入力の符号部を前記第 2 の形式の浮動小数点データの符号部とし、前記第 1 の形式の浮動小数点データ入力の指数部を前記第 2 の形式の浮動小数点データの指数部とし、前記第 1 の形式の浮動小数点データ入力の仮数部を、前記m1 ビットの前記第 2 の形式の浮動小数点データの仮数部のうちの上位m0 ビットとすると共に、前記第 2 の形式の浮動小数点データの仮数部の下位m1 - m0 ビットを所定の数値として、前記第 1 の形式の浮動小数点データ入力を、前記第 2 の形式の浮動小数点データに変換する入力データ変換工程と、

前記入力データ変換工程で変換された前記第 2 の形式の浮動小数点データについて、浮動小数点演算を実行し、その演算結果としての前記第 2 の形式の浮動小数点データを得る浮動小数点演算工程と、

前記浮動小数点演算工程で得られる前記演算結果としての前記第 2 の形式の浮動小数点データの符号部を前記第 1 の形式の浮動小数点データの符号部とし、前記演算結果としての前記第 2 の形式の浮動小数点データの指数部を前記第 1 の形式の浮動小数点データの指数部とし、さらに前記演算結果としての前記第 2 の形式の浮動小数点データの仮数部の上位m0 ビットを前記第 1 の形式の浮動小数点データの仮数部として、前記演算結果としての前記第 2 の形式の浮動小数点データを、前記第 1 の形式の浮動小数点データ出力に変換して、前記バスに出力する出力データ変換工程と、

を備えることを特徴とする並列計算方法。

【請求項 7】ホストプロセッサと、このホストプロセッサとバスを介して接続される 1 または複数のプロセッサエレメントとにより、特定の計算プロセスに含まれる浮動小数点演算処理を分担して計算する並列計算方法において、

前記ホストプロセッサは、

1 ビットの符号部と、me ビットのビット幅の指数部と、m0 ビットのビット幅の仮数部とからなる第 1 の形式の浮動小数点データについて演算処理を行い、

4

前記プロセッサエレメントは、

前記第 1 の形式よりも高精度で、1 ビットの符号部と、前記me ビットのビット幅の指数部と、前記m0 よりも大きいm1 ビットのビット幅の仮数部とからなる第 2 の形式の浮動小数点データについて、浮動小数点演算処理を実行するものであって、かつ、

前記バスを通じて入力される第 1 の形式の浮動小数点データ入力の符号部を前記第 2 の形式の浮動小数点データの符号部とし、前記第 1 の形式の浮動小数点データ入力の指数部を前記第 2 の形式の浮動小数点データの指数部とし、さらにm1 ビットの第 2 の形式の浮動小数点データの仮数部のうち、その最上位 1 ビットは 1 とし、最上位よりも下位の前記m0 ビットは前記第 1 の形式の浮動小数点データ入力の仮数部とし、さらにそれより下位の前記m1 - 前記m0 - 1 ビットは所定の数値として、前記第 1 の形式の浮動小数点データ入力を、前記第 2 の形式の浮動小数点データに変換する入力データ変換工程と、

前記入力データ変換工程で変換されて得られた前記第 2 の形式の浮動小数点データについて、浮動小数点演算を実行して、その演算結果としての前記第 2 の形式の浮動小数点データを得る浮動小数点演算工程と、

前記浮動小数点演算工程で得られる前記演算結果としての前記第 2 の形式の浮動小数点データの符号部を前記第 1 の形式の浮動小数点データの符号部とし、また前記演算結果としての前記第 2 の形式の浮動小数点データの指数部を前記第 1 の形式の浮動小数点データの指数部とし、さらに前記演算結果としての前記第 2 の形式の浮動小数点データの仮数部の上位 2 ビット目からのm0 ビットを、m0 ビットの前記第 1 の形式の浮動小数点データの仮数部として、前記演算結果としての前記第 2 の形式の浮動小数点データを、前記第 1 の形式の浮動小数点データ出力に変換して、前記バスに出力する出力データ変換工程と、を備えることを特徴とする並列計算方法。

【請求項 8】前記計算プロセスは、分子軌道法に基づく計算プロセスであることを特徴とする請求項 1 ~ 5 のいずれかに記載の並列計算装置。

【請求項 9】前記計算プロセスは、分子軌道法に基づく計算プロセスであることを特徴とする請求項 6 または請求項 7 に記載の並列計算方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】この発明は、医薬品の分子構造設計や物性の予測などに用いられる非経験的分子軌道計算を初めとする、浮動小数点演算による高精度な積和演算を多数含んだ物理現象の解析に使用される並列計算システムに関する。

【0002】

【従来の技術】近年、半導体の微細化、高速化に伴い、計算機の高性能化が実現され、薬学における分子構造設

10

20

30

40

50

5

計、物性値予測の分野にも、非経験的分子軌道計算を用いた分子シミュレーションが行われるようになってきた。

【0003】非経験的分子軌道計算の中でも、比較的計算量が少なく、定性的な解析には十分に対応することができる、ハートレーフォック法(HF法)が最も広く用いられている。このHF法については、例えば「藤永茂、分子軌道法、岩波書店(1980)」、「大澤映二編、分子軌道法、講談社サイエンティフィク(1994)」、「菊池修、基礎量子化学、朝倉書店(1997)」などに示されている。以下、HF法の概要について述べる。

【0004】HF法は、フォック方程式を、後述するSCF法によって解く方法として定式化されている。ここで、フォック方程式は、分子に含まれている原子軌道の総数をN、原子軌道の線形近似で表される分子軌道の総

$$\begin{aligned} F r s &= h r s + g r s \\ &= h r s + \sum [t, u = 1 \sim N] P t u ((r s, t u) \\ &\quad - (1/2) (r t, s u)) \end{aligned} \quad (2)$$

【0008】この(2)式の $h r s$ は、1電子に対するエネルギーを表す積分量であり、(1)式の1回の計算で N^2 個に比例する数だけ計算される。

【0009】なお、この明細書において、 $\sum [i, j = 1 \sim N] f(i, j)$ は、 i および j について1からNまでの総和を関数 $f(i, j)$ について求める演算を示

$$P t u = \sum (j = 1 \sim m) C t j \cdot C u j \quad (3)$$

また、(2)式の $(r s, t u)$ ($r, s, t, u = 1 \sim N$)は2電子積分と呼ばれる物理量であり、原子軌道 $\chi_i(r)$ ($i = 1 \sim N$, r は座標)を用いて以下の式

$$\begin{aligned} (r s, t u) &= \int \int \chi_r(r_1) \chi_s(r_1) (1/r_{12}) \\ &\quad \times \chi_t(r_2) \chi_u(r_2) dr_1 \cdot dr_2 \end{aligned} \quad (4)$$

【0013】ここで、 r_1, r_2 はそれぞれ独立した2つの座標系であり、それぞれ全空間にわたって二重積分が行われる。また、 r_{12} は、座標系 r_1 と r_2 との間の距離を表す。この2電子積分は、 r, s, t, u が、それぞれ原子軌道の数だけ存在するので、(1)式の1回

$$S r s = \int \chi_r(r_1) \chi_s(r_1) dr_1 \quad (5)$$

このように表されるので、HF法は(1)式で表される m 個の固有値 ϵ_i 、固有ベクトル C_i ($i = 1 \sim m$)を求める問題となる。しかしながら、(2)式、(3)式より分かるように、(1)式に含まれるフォック行列は、係数を表すベクトル C_i を使って求められるので、(1)式を解いて得られる C_i を使用しないと、 F の値も求められないことになる。

【0016】したがって、まず、 C_i の初期推定値(initial guess)として適当な値を設定し、その C_i を使用して F を求め、(1)式の固有値問題を解いて、新たな C_i を求める。次に、この C_i を使って、新たな F を計算して(1)式を解く。このように繰り返し計算を行い、最後に F の計算に使用された C_i

6

数を m とすると、分子全体に関するシュレディンガー方程式に対して1電子近似、線形近似を行った結果得られる

$$F C = S C \epsilon \quad (1)$$

という式で表される。このフォック方程式を解くことによって、分子の有するエネルギーが求まるため、その値により分子が安定な状態かどうかを判定できる。

【0005】(1)式において、 F はフォック行列と呼ばれる $N \times N$ の行列であり、また、 S は重なり行列と呼ばれる $N \times N$ の行列、 C は係数を表す $N \times m$ の行列、 ϵ は分子軌道を占有するそれぞれの電子が持つエネルギーを表す $m \times m$ の対角行列である。

【0006】ここで、フォック行列の要素 $F r s$ ($r, s = 1 \sim N$)は、以下の式で表される。

【0007】

すものとしている。また、 $\sum [i = 1 \sim N] f(i)$ は、 i について1からNまでの総和を関数 $f(i)$ について求める演算を示すものとしている。

【0010】(2)式の $P t u$ は、密度行列と呼ばれ、以下のように上記の行列 C を用いて表される。

$$P t u = \sum (j = 1 \sim m) C t j \cdot C u j \quad (3)$$

のように表される。

【0012】

の計算で N^4 個に比例した数だけ必要となる。

【0014】次に、重なり行列 S の要素 $S r s$ は以下の式で表される。

【0015】

$$S r s = \int \chi_r(r_1) \chi_s(r_1) dr_1 \quad (5)$$

と、求められた C_i との間に殆ど差がなくなったところで計算を終了する。この方法は、SCF(self-consistent field)法と呼ばれ、分子軌道計算において広く使われている方法である。

【0017】(1)式で表される2電子積分の個数は、原子軌道の総数 N の4乗に比例するため、例えば生物学などの分野でよく現れる100個程度の原子からなる分子を考えた場合、 N の値は1000程度となり、その4乗の100兆個のオーダーにものぼる。ここで、2電子積分を計算する前に、値が小さいものを判定してカットオフする方法が良く用いられるものの、計算が必要な2電子積分の数は1億個程度であり、膨大な数であることに変わりはない。

【0018】このため、SCF法の各反復には同じ2電子積分が使用されるものの、2電子積分を一旦計算して格納しておくメモリスペースがないため、反復毎に2電子積分を計算し直すというダイレクト法が通常用いられる。このダイレクト法による分子軌道計算では、2電子積分の計算に大部分の計算時間が占有されるため、この部分の高速化が重要となる。

【0019】ここで、(4)式で表される原子軌道 χ_i には、通常、2電子積分を解析的に求めることができるガウス型関数を使用される。このガウス型関数の原子軌道を用いた高速な2電子積分の計算法としては、従来、文献1” S. Obara and A. Saika, J. Chem. Phys. 84, 3963 (1986)”に示されている方法(以下、小原の方法と称す

$$\chi(r-R; n, \zeta) = (r_x - R_x)^{n_x} (r_y - R_y)^{n_y} \times (r_z - R_z)^{n_z} \exp[-\zeta(r-R)^2] \quad (6)$$

【0023】ここで、 r と R は、空間上の位置を表すベクトルであり、特に、 R は原子の中心を表す。また、 n は0以上の整数で構成されるベクトルであり、軌道量子数ベクトルと呼ばれる。この軌道量子数ベクトルは、 r

定数である。また、軌道量子数ベクトルの3成分の和

$$\lambda = n_x + n_y + n_z \quad (7)$$

は、軌道量子数と呼ばれる。軌道量子数 λ が、それぞれ0、1、2、3のとき、対応するガウス関数はs関数、p関数、d関数、f関数と呼ばれ、原子の持つs軌道、p軌道、d軌道、f軌道にそれぞれ対応した波動関数として扱われる。

【0025】各軌道に対する波動関数は、これらの関数の線形結合で近似される。例えば水素原子の1s軌道に対応する波動関数は、(7)式において $n = (0, 0, 0)$ とした上で、軌道指数 ζ が数種類の異なる値をとるものの線形結合で表される。このときの ζ の種類は基底関数系で異なる。

【0026】なお、軌道量子数 λ が0でないときには、波動関数は複数個存在する。例えば、軌道量子数 λ が2で表されるd関数は、軌道量子数ベクトル $n = (2, 0, 0)$ 、 $(0, 2, 0)$ 、 $(0, 0, 2)$ 、 $(1, 1, 0)$ 、 $(0, 1, 1)$ 、 $(1, 0, 1)$ に応じて、 dx^2 、 dy^2 、 dz^2 、 dxy 、 dyz 、 dzx の6通りが存在する。

【0027】したがって、(4)式で表される2電子積分の中に1個のd関数が含まれている場合は、これらの関数に応じて6通りの2電子積分が求められる。こ

る)が知られていた。

【0020】小原の方法は、2電子積分を拡張した補助積分という値を導入し、補助積分を含んだ漸化式の形式で表される。この漸化式によって、1つの2電子積分は、より低次の補助積分を含む積和演算の形式によって表される。ある2電子積分を求めるときは、まず、漸化式に従って、最も次数の低い補助積分だけを含んだ形式に展開し、次に、積和演算によって、次数の高い補助積分を順次求めていくことによって計算が行われる。以下に、小原の方法の具体的な計算方法を示す。

【0021】まず、小原の方法では、ガウス型関数で表される原子軌道 χ は、以下の式で表されている。

【0022】

などと同様、 x 、 y 、 z の3つの成分 n_x 、 n_y 、 n_z をもつ。

【0024】さらに、 ζ は、原子の種類や軌道の種類に応じて変化する軌道指数と呼ばれる

で、(6)式で表されるガウス型関数において、軌道量子数と、原子の中心座標 R と、パラメータ ζ とが同じガウス型関数の集合は、シェルと呼ばれる。したがって、軌道量子数 λ に応じてpシェル、dシェルなどに分類され、例えばdシェルには6つのガウス型関数が含まれている。

【0028】なお、(4)式の左辺で表される2電子積分は、 $(p * s, p * s)$ など、それぞれの波動関数が有するp、sなどの型で表すこともある。この場合は、型の名前には「*」を付与して、波動関数名と区別するものとする。

【0029】前記文献1では、(1)式で表される2電子積分に含まれる波動関数a、b、c、dを、(3)式で表現されるガウス型関数を使って表わしたとき、2電子積分の値を効率的に計算する方法が、再帰的な式で表されている。小原の方法では、2電子積分(a b, c d)に対して補助積分(a b, c d)^(m) (mは0以上の整数)という補助的な物理量が導入され、これを使って以下のような形式の再帰式が導かれている。

【0030】

$$\begin{aligned} (a b, c d) &= (a b, c d)^{(0)} \quad (8) \\ (a + 1 i b, c d)^{(m)} &= P 0 (a b, c d)^{(m)} \\ &\quad + P 1 (a b, c d)^{(m+1)} \\ &\quad + N i (a) \times P 2 (a - 1 i b, c d)^{(m)} \\ &\quad + N i (a) \times P 3 (a - 1 i b, c d)^{(m+1)} \\ &\quad + N i (b) \times P 4 (a b - 1 i, c d)^{(m)} \\ &\quad + N i (b) \times P 5 (a b - 1 i, c d)^{(m+1)} \end{aligned}$$

$$+N_i(c) \times P_6(a, b, c-1, d)^{(m+1)} \\ +N_i(d) \times P_7(a, b, c, d-1, i)^{(m+1)} \quad (9)$$

(但し、 $i=x, y, z$)

$$(a, b, c, d)^{(m)} = (s * s *, s * s *)^{(m)} \\ = P_8 \times K(A, B, \zeta a, \zeta b) \times K(C, D, \zeta c, \zeta d) \times F_m(T)$$

(10)

【0031】ここで、波動関数 a, b, c, d は、全て (6) 式で表されるガウス型関数であり、それぞれが特定の軌道量子数ベクトル n を持つ。また、記号 $a+1, i$ は、ガウス型関数 a の軌道量子数ベクトルのうち、 i 成分 ($i=x, y, z$) の値を1つだけ増加させたガウス型関数を意味する。したがって、例えば、 a が $n=(1, 0, 0)$ で表される p_x 関数の場合、 $a+1, x$ は、 $n=(2, 0, 0)$ で表される d_{x^2} 関数となる。

【0032】また、記号 $a-1, i$ は、ガウス型関数 a の軌道量子数ベクトルのうち、 i 成分 ($i=x, y, z$) の値を1つだけ減少させたガウス型関数を意味する。また、記号 $N_i(a)$ は、ガウス型関数 a の軌道量子数ベクトルの i 成分を表す。したがって、例えば波動関数 b の軌道量子数ベクトルの i 成分が0であるときは、

(9) 式の $b-1, i$ に関する項は0となる。

【0033】また、補助積分に対して成立する関係

$$(a, b, c, d)^{(a)} = (b, a, c, d)^{(a)} \\ = (c, d, a, b)^{(a)} \\ = (d, c, a, b)^{(a)}$$

を利用すると、 $(a, b+1, i, c, d)^{(m)}$ 、 $(a, b, c+1, i, d)^{(m)}$ 、 $(a, b, c, d+1, i)^{(m)}$ に関しても (9) 式と同様の関係式を導くことができる。

【0034】さらに、(9) 式の補助積分の係数 $P_0 \sim P_7$ は、中心となる原子核の座標 A, B, C, D や波動関数 a, b, c, d の (6) 式の軌道指数 ζ に対応する

$$P_8 = (\zeta + \eta)^{-1/2} \quad (25)$$

$$T = \rho (P - Q)^2 \quad (26)$$

$$K(R_0, R_1, \zeta_0, \zeta_1) = \{2^{1/2} \cdot \pi^{5/4} / (\zeta_0 + \zeta_1)\} \\ \times \exp[-\{\zeta_0 \zeta_1 / (\zeta_0 + \zeta_1)\} (R_0 - R_1)^2]$$

(27)

$$F_m(T) = (0 \sim 1) \int t^{2m} \cdot \exp[-T t^2] dt$$

(28)

【0038】ここで、(28) 式の関数 $F_m(T)$ は誤差関数と呼ばれ、上記の文献1ではテーラー展開を使って計算する方法が挙げられている。なお、(28) 式において、 $(0 \sim 1) \int$ は、0から1までの有限積分を示すものとする。

【0039】以上のように、小原の方法では2電子積分が(8)～(10)式のように再帰的に表わされているので、(9) 式を繰り返して適用して、右辺に軌道量子数が0の補助積分が現れるようにして、さらに、(10) 式を用いて軌道量子数が0の補助積分を求めることによって、目的とする2電子積分の値を得ることができる。

$\zeta a, \zeta b, \zeta c, \zeta d$ から計算されるパラメータであり、以下の式で表される。

【0035】

$$P_0 = P_i - A_i \quad (11)$$

$$P_1 = W_i - P_i \quad (12)$$

$$P_2 = 1 / (2 \zeta) \quad (13)$$

$$P_3 = -\rho / (2 \zeta^2) \quad (14)$$

$$P_4 = 1 / (2 \zeta) \quad (15)$$

$$P_5 = -\rho / (2 \zeta^2) \quad (16)$$

$$P_6 = 1 / (2 (\zeta + \eta)) \quad (17)$$

$$P_7 = 1 / (2 (\zeta + \eta)) \quad (18)$$

ここで、

$$\zeta = \zeta a + \zeta b \quad (19)$$

$$\eta = \zeta c + \zeta d \quad (20)$$

$$P = (\zeta a A + \zeta b B) / \zeta \quad (21)$$

$$Q = (\zeta c C + \zeta d D) / \eta \quad (22)$$

$$W = (\zeta P + \eta Q) / (\zeta + \eta) \quad (23)$$

$$\rho = \zeta \eta / (\zeta + \eta) \quad (24)$$

となる。

【0036】また、(10) 式の右辺を形成する係数 P_8 、パラメータ T 、および関数 $K(R_0, R_1, \zeta_0, \zeta_1)$ 、関数 $F_m(T)$ は、それぞれ以下の関係式で表される。

【0037】

【0040】例えば、2電子積分 ($p * p *, s * s *$) を求める場合を例にして説明する。ここで、2電子積分に含まれる4つのガウス型関数は、すべて特定のシェルに含まれているとする。このとき、前述のように、 p 関数には、その軌道量子数ベクトルに応じて、 p_x, p_y, p_z の3通りが存在するから、求めるべき2電子積分は、($p_x * p_x *, s * s *$)、($p_x * p_y *, s * s *$)、($p_x * p_z *, s * s *$)、($p_y * p_x *, s * s *$)、($p_y * p_y *, s * s *$)、($p_y * p_z *, s * s *$)、($p_z * p_x *, s * s *$)、($p_z * p_y *, s * s *$)、($p_z * p_z *, s * s *$) の9通りとなる。

【0041】ここで、まず、このうちの $(p x * p y * , s * s *)$ を求める場合について説明する。

(8)、(9) 式によって展開すると、以下の関係が成

$$\begin{aligned} & (p x * p y * , s * s *) \\ &= (p x * p y * , s * s *)^{(0)} \\ &= P 0 (s * p y * , s * s *)^{(0)} \\ &+ P 1 (s * p y * , s * s *)^{(1)} \end{aligned} \quad (29)$$

【0043】この(29)式の右辺の補助積分は、2番目の波動関数だけが s 関数でない形になっている。この形式の補助積分に対しても、前述したように、波動関数の入れ替えによって、(9)式と同様の式を得ることが

$$\begin{aligned} & (s * p y * , s * s *)^{(0)} \\ &= P 0' (s * s * , s * s *)^{(0)} \\ &+ P 1' (s * s * , s * s *)^{(1)} \\ & (s * p y * , s * s *)^{(1)} \\ &= P 0'' (s * s * , s * s *)^{(1)} \\ &+ P 1'' (s * s * , s * s *)^{(2)} \end{aligned} \quad (30)$$

【0045】このようにして、全ての項を $(s * s * , s * s *)^{(m)}$ の形で表すことができた。次に、(30)式、(31)式の右辺の $(s * s * , s * s *)^{(0)}$ 、 $(s * s * , s * s *)^{(1)}$ 、 $(s * s * , s * s *)^{(2)}$ を、(10)式を使用して求め、以後、展開とは逆の順番で、最初に求められた値を、(30)式、(31)式の右辺に代入して、 $(s * p y * , s * s *)^{(0)}$ と $(s * p y * , s * s *)^{(1)}$ を求める。さらに、これを(29)式の右辺に適用して最終的に $(p x * p y * , s * s *) = (p x * p y * , s * s *)^{(0)}$ を得ることができる。

【0046】また、残りの8つの2電子積分も、(29)式、(30)式、(31)式と同様の数式に展開して求めることができる。このとき、(30)式、(31)式で使用した $(s * s * , s * s *)^{(0)}$ 、 $(s * s * , s * s *)^{(1)}$ 、 $(s * s * , s * s *)^{(2)}$ が必ず使用されるので、これらの値を求め直すことなく、再利用できる。また、これらの2電子積分は、原子核の座標と s の値が同じであるため、積和計算に使用する $P 0$ 、 $P 1$ などの係数の値も、ほぼ共通に使用することができる。

【0047】このように、2電子積分に含まれる4つのガウス型関数が、それぞれ特定のシェルに含まれているときは、共通に利用できる補助積分が多い。したがって、特定のシェルに含まれているガウス型関数を使った2電子積分は、通常、まとめて計算される。2電子積分の値は、以上の手順に従って、(8)式～(10)式を使用することによって求めることができる。

【0048】以上述べたように、2電子積分の計算は、(9)式のように、(係数×補助積分)の値を、別の補助積分の値に足し込んでいく、浮動小数点積和演算の繰り返しで表される。このとき、(28)式の誤差関数の

立する。

【0042】

できる。したがって、(29)式の右辺の補助積分を、(9)式に再度適用して展開することによって、以下の2つの式を得ることができる。

【0044】

(31)

計算は、テーラー展開で表され、さらに(13)式などに現れる逆数、(25)式に現れる平方根の逆数、さらに(27)式に現れる \exp 計算も、公知のニュートン法、テーラー展開で計算できる。

【0049】ニュートン法やテーラー展開も、浮動小数点積和演算の繰り返しによって表わすことができるので、結局、2電子積分は、連続する積和演算の繰返しとなる。また、(9)式による1回の展開によって最大8個の積和演算が増加するため、1つの2電子積分を計算するときに必要となる積和演算の数も多くなる。

【0050】以上述べたように、HF法に従った分子軌道計算を行う場合、SCF計算の1回の繰返しにおいて現れる2電子積分を、浮動小数点積和演算によって計算することが必要とされる。この計算量は、2電子積分の数および1つの2電子積分の計算に要する計算量の観点から見て、膨大なものとなる。

【0051】従来、分子軌道計算を高速に行う例として、「超高速分子軌道計算専用機MOEのアーキテクチャ」(白川他、信学技報、CPSY96-46(1996-05)) (文献2)があった。これは、2電子積分の計算は、ある程度独立に可能であることを利用して、複数の2電子積分の計算を並列化するものである。

【0052】この文献2の例では、システムに、複数のプロセッサエレメントを用意して、それぞれのプロセッサエレメントに、2電子積分の計算の一部を分担させる。2電子積分の計算は、前述の小原の方法に基づく浮動小数点積和演算によって行われ、プロセッサエレメントは、(2)式で表されるフォック行列のうち、 $g r s$ までを計算して、この値をホストプロセッサに送信する。その後、ホストプロセッサでは、(1)式で表される固有値問題が解かれる。

【0053】また、別の例としては、特開平9-504

28号公報に示されているものがある。この例では、複数の相互接続された計算機よりなる計算機クラスタを備え、各計算機において、2電子積分や(2)式で表されるフォック行列要素を計算し、この値をベクトル計算機に送る。その後、ベクトル計算機において(1)式で表される固有値問題が解かれる。

【0054】

【発明が解決しようとする課題】以上述べたような2電子積分やフォック行列要素の計算では、前述の通り膨大な浮動小数点演算が行われるため、演算の途中で発生する誤差およびそれによる精度が問題となってくる。そこで、まず、計算に必要とされる精度について検討する。

【0055】分子軌道計算の規模は、分子の大きさや、使用する基底系によっても変わってくるため、計算に必要とされる精度を一律に決めることは困難であるが、前述の文献1によれば、非経験的分子軌道計算の結果が十分な精度をもつための目安として、(28)式で表される誤差関数の値が持つ相対誤差が 10^{-15} 程度よりも小さい精度を持つことが必要とされている。

【0056】つまり、誤差関数を、この程度の精度で計算することによって、最終的にその誤差関数を使った分子軌道計算で得られるエネルギーの値が、特定の範囲の規模の分子軌道計算を行う場合に対して、実用上十分な精度となる。そこで、文献1に示された目安をもとにして、精度の見積もりを行うことにする。

【0057】ここで、IEEE規格754で定められている倍精度の浮動小数点表現では、その仮数部の長さが52ビットであり、このうち、最上位の1の値は含まれないために、53ビットの精度をもつと考えられる。このビット長は、相対誤差にして、 $2^{-53} = 1.11 \times 10^{-16}$ であるから、倍精度の浮動小数点は、上記精度を満足している。

【0058】そこで、ここでは、(28)式に従って求められた誤差関数が、仮数部が53ビットの倍精度浮動小数点の精度を持っていることを条件として、積和演算が満足すべき精度を見積もることとする。

【0059】このとき、前に述べたように、(21)～(24)式などで用いられる割り算の計算には、ニュートン法による逆数計算を用い、(28)式の誤差関数の計算には、テーラー展開による計算を用いることを前提とする。したがって、誤差関数の値は、予め与えられたA、B、C、D、 ζa 、 ζb 、 ζb 、 ζd の値から積和演算の繰り返しで計算する。

【0060】数学辞典(岩波書店)などによると、一般に、誤差は、(1)入力誤差、(2)打ち切り誤差、

(3)丸め誤差、の3つに分類される。誤差関数の計算値には、これら3種類の誤差が影響すると考えられる。そこで、積和演算に必要な精度を見積もる準備として、

$$A + rB = (1 + \epsilon)(A + iB)$$

$$A \times rB = (1 + \epsilon)(A \times iB)$$

これら3つの誤差による影響について説明する。

【0061】まず、入力誤差とは、予め与えられるデータの中に既に存在している誤差のことであり、浮動小数点で表されたA、B、C、D、 ζa 、 ζb 、 ζb 、 ζd (初期パラメータと称する)に含まれる誤差のことを指す。ここでは、初期パラメータは、倍精度浮動小数点で与えられると仮定する。したがって、これらの値は53ビット仮数部以上の精度を持っており、入力誤差もこの程度である。

【0062】次に、打ち切り誤差とは、例えば上述のように割り算をニュートン法、また、(28)式の値をテーラー展開で近似したときに、その繰り返し回数や展開項数を有限回数で打ち切ることによって由来する誤差である。この誤差は、ニュートン法の繰り返し数、テーラー展開の展開項数や、初期値や係数値の値の精度を大きくとることによって制御できる。したがって、打ち切り誤差も入力誤差と同様十分に小さくするように制御すると仮定する。

【0063】さらに、丸め誤差とは、計算の各段階で得られる数値を限られた桁数に丸めるために発生する誤差のことである。ここでは、有限長の乗算および加減算による積和演算を繰り返し用いることによって、丸め誤差が発生する。

【0064】このとき、入力誤差、打ち切り誤差とも十分に小さく、倍精度浮動小数点のもつ誤差以下と仮定しているため、これら2種類の誤差が誤差関数の値に与える影響は少ない。そこで、この解析では丸め誤差だけを考慮し、積和演算の繰り返しによって累積する丸め誤差を考慮しても、計算された誤差関数の値が、なお、53ビット仮数部を持つ倍精度浮動小数点の精度を維持している、という条件から、積和演算に必要な精度を計算できる。

【0065】次に、誤差関数の精度を具体的に解析する前に、誤差関数の計算で使用される要素的な計算の誤差を見積もる。

【0066】まず、積和演算に含まれる加算、乗算によって発生する相対誤差について説明する。この説明では、有限長の桁数で行われて丸め誤差が生じる加算、乗算を便宜的に $+r$ 、 $\times r$ と表記し、無限長の桁数で行われて丸め誤差が発生しない加算、乗算を便宜的に $+i$ 、 $\times i$ と表記することとする。また、簡単のため、有限長の加算、乗算によって、一律に、有限長演算によって発生する最悪の相対誤差 ϵ だけ相対誤差が発生するものと仮定する。

【0067】このとき、あるパラメータA、Bの間の有限長の加算、乗算は、以下の(32)式、(33)式のように表される。

【0068】

$$(32)$$

$$(33)$$

【0069】また、パラメータA、B、C、Dを使っ

$$X = A + rB$$

$$Y = C + rD$$

と表されるとき、XとYとの有限長の加算、乗算は以下の(36)式、(37)式のように表される。

$$\begin{aligned} X + rY &= (A + rB) + r(C + rD) \\ &= (1 + \varepsilon)(A + iB) + r(1 + \varepsilon)(C + iD) \\ &= (1 + \varepsilon)^2((A + iB) + i(C + iD)) \end{aligned} \quad (36)$$

$$\begin{aligned} X \times rY &= (A + rB) \times r(C + rD) \\ &= (1 + \varepsilon)(A + iB) \times r(1 + \varepsilon)(C + iD) \\ &= (1 + \varepsilon)^3((A + iB) \times r(C + iD)) \end{aligned} \quad (37)$$

【0071】したがって、(36)式のように有限長の加算を行うときは、理想的な結果に対して相対誤差の2乗がかかり、(37)式のように有限長の乗算を行うときは、理想的な結果に対して相対誤差が3乗でかってくる。このように、丸め誤差を含んだ式同志の演算では、式に含まれている相対誤差の残り方が、加算の場合と乗算の場合とで異なる。

$$\begin{aligned} &((A + rB) + rC) + rD \\ &= (1 + \varepsilon)^3 A + i(1 + \varepsilon)^3 B + i(1 + \varepsilon)^2 C + i(1 + \varepsilon) D \end{aligned} \quad (38)$$

【0074】したがって、AとBに対しては、相対誤差が $(1 + \varepsilon)^3$ になるため、この式全体に関しても、理想的な結果に対して相対誤差が $(1 + \varepsilon)^3$ になるとみなされる。このように、加算の順番に対しても丸め誤差が変わってくる。

【0075】次に、逆数の値を求めるニュートン法による丸め誤差について解析する。ある値yの逆数を求めるためのニュートン法のステップは、まず、値yを特定の範囲に正規化して、その逆数に近いxの値を選び、以下の(39)～(41)式を繰り返すことによって得られるxの計算値を、逆に正規化することで得ることができる。

【0076】

$$y_0 = y \times x \quad (39)$$

$$r = 2 - y_0 \quad (40)$$

$$x = r \times x \quad (41)$$

【0077】このときの繰り返し回数は、最初に選択したxの値と真の値との差に依存するので、初期値の正確さに依存する。ここでは、繰り返し回数を3としたときに、十分な打ち切り誤差となると仮定する。このとき、

(39)～(41)式を有限長の乗算、加算で行うことによって、相対誤差は、それぞれ $(1 + \varepsilon)$ が1回ずつかかる形になる。したがって、3回の繰り返しによって

$$F_6 = (1/6) \times (T - T') \times C_6 \quad (43)$$

$$F_5 = (1/5) \times (T - T') \times (F_6 + C_5) \quad (44)$$

$$F_4 = (1/4) \times (T - T') \times (F_5 + C_4) \quad (45)$$

.....

$$F_m(T) = F_0 = (T - T') \times (F_1 + C_0) \quad (46)$$

【0082】(43)～(46)式を有限長の乗算、加算を使って行うとき、(43)式にかかる相対誤差は

て、パラメータX、Yが

$$(34)$$

$$(35)$$

【0070】

【0072】さらに、パラメータA～Dの累積を求める場合、(36)式のように、X、Yを、まず、求めてから両者を加算する方法では、相対誤差が $(1 + \varepsilon)^2$ になるのに対し、パラメータA～Dを順番に足していく方法では、その式は、以下の(38)式に示すように展開される。

【0073】

発生する相対誤差は $(1 + \varepsilon)^9$ となる。

【0078】ニュートン法は、上記(39)～(41)式の繰り返しによってxの値を $(1/y)$ に近づけていく方法であるため、実際には、繰り返しの途中で混入する丸め誤差も補償されていく。このため、上記の $(1 + \varepsilon)^9$ という相対誤差の値は、オーバーエスティメイトであるが、ここでは最悪のケースを想定して、この値を使用することにする。

【0079】さらに、(28)式に対し、テーラー展開を適用したときの丸め誤差について解析する。前述の文献1には、7項の展開の例が示されているので、ここでも7項の展開を仮定する。このとき、テーラー展開は、以下の(42)式で表される。

【0080】 $F_m(T) = \sum [k=0 \sim 6] 1/k! \cdot C_k (T - T')$ (42) ここで、 T' は、与えられたTの範囲を所定の分割数で分割したときに、Tに最も近い境界値である。この分割数は、前述のように、テーラー展開の打ち切り誤差が小さくなるように、十分大きく取られていると仮定する。また、 C_k はその T' の値に応じて決定される係数の値である。このテーラー展開は、実際には以下のような手順で計算される。

【0081】

30

40

50

$(1+\varepsilon)^3$ 、また、(44)式以降にかかる相対誤差は $(1+\varepsilon)^4$ となる。但し、F4、F2、F1、F0に対しては、最初の(整数分の1)の乗算は正確にできるので、 $(1+\varepsilon)^3$ となる。この結果、テラー展開によって発生する相対誤差は、 $(1+\varepsilon)^{24}$ となる。

【0083】実際のテラー展開では、分割数は上述のように十分に大きくとられるため、(42)式の $(T-T')$ の値は十分に小さい。したがって、例えば(44)式に含まれる $(F6+C5)$ のように、直前に求められた値Fと係数Cとの足し算では、直前に $(1/k)$ や $(T-T')$ の値が掛け合わされているFよりも係数Cの方が十分大きく、Fの値に混入している丸め誤差の値は問題とならない可能性もある。このため、上記 $(1+\varepsilon)^{24}$ という相対誤差の値は、オーバーエスティメイトであるが、最悪のケースを想定して、この値を使用することにする。

【0084】さらに、(43)～(46)式によって求められる際に、 $(T-T')$ が1回ずつ掛け合わされる。このため、機械的に解析すると、Tに含まれる相対誤差は6乗となると思われるが、この相対誤差も繰り返しの各段階におけるFの値に混入していくため、上で述べた理由により6乗とすることもオーバーエスティメイトである。しかし、ここでも最悪の場合を想定してTに含まれる誤差を6乗とみなすことにする。

【0085】次に、以上述べた誤差関数の要素的な計算に混入する相対誤差を考慮しながら、誤差関数の計算に現れる丸め誤差について述べる。上述したように、ここで見積もった丸め誤差の値は、オーバーエスティメイトな部分も含まれているが、ここでは誤差関数の値が所定の誤差の範囲内にとどまるようにするために十分となる積和演算の精度を求めたいので、見積もった誤差の値をそのまま使用する。

【0086】図7は、初期パラメータA、B、C、D、 ζa 、 ζb 、 ζc 、 ζd から誤差関数 $F_m(T)$ を計算するまでの経路と、各計算で発生する相対誤差とを示したものである。この図7では、誤差関数が初期パラメータから順次積和演算を使って求められる経路を矢印で示し、各積和演算だけによってかかる相対誤差の値を式の右側に示している。

【0087】まず、(26)式のTの値に含まれる相対誤差を検討する。ここで、Tは、 ρ と、 $(P-Q)^2$ との2つの値の乗算で表されているため、これら2つの値に含まれる丸め誤差について述べる。

【0088】最初に、 $(P-Q)^2$ の相対誤差は、図7

$$(1+\varepsilon)^{294} < (1+2^{-53})$$

ここで、 ε が 2^{-53} より小さい数字であるため、 $\varepsilon < 2^{-53}$ である。この結果、(47)式の左辺は $1+2^{-53}$ と近似でき、最終的に、(47)式は、 $\varepsilon \leq 2^{-53}$ と変形される。この条件式より、誤差関数が倍精度浮動小数点程度の精度を持つために十分となる積和演算器の

のステップ1～5のパスを解析することによって得られる。ステップ1～3までは、各ステップでの相対誤差をそのまま掛け合わせることで、Pxの相対誤差が得られ、その値は $(1+\varepsilon)^{13}$ である。

【0089】次に、ステップ4の値は、Pxを含む項が2回かかった形になっているため、相対誤差は、 $((1+\varepsilon)^{13})^2 \times (1+\varepsilon)^3 = (1+\varepsilon)^{29}$ とみなし、さらに、ステップ5の相対誤差をかけると、 $(1+\varepsilon)^{31}$ となる。この値が、 $(P-Q)^2$ の相対誤差となる。

【0090】次に、 ρ の相対誤差は、ステップ1、6、7、8のパスと、ステップ1、8のパスを考慮する。ステップ1、6、7で、相対誤差は、 $(1+\varepsilon)^{11}$ であり、ステップ8では、3者の乗算となるため、結局 $(1+\varepsilon)^{13}$ となる。この値が、 ρ の相対誤差となる。

【0091】以上求められた、 $(P-Q)^2$ の相対誤差、 ρ の相対誤差から、ステップ9で求められるTに含まれる相対誤差は、 $(1+\varepsilon)^{31+13+1} = (1+\varepsilon)^{45}$ と計算される。

【0092】さらに、Tに含まれる相対誤差から、ステップ10の誤差関数に含まれる相対誤差を求める。前述したテラー展開の丸め誤差による影響に基づいて計算すると、誤差関数の相対誤差は、 $(1+\varepsilon)^{45 \times 6 + 24} = (1+\varepsilon)^{294}$ となる。

【0093】実際の誤差解析では、減算にともなう桁落ちも考慮しなければならない。この場合は、(26)式に含まれる $(P-Q)$ の計算において、Pの値とQの値がかなり近い場合に、減算結果の有効桁数が減少してしまうものである。ただし、この場合は、 $(P-Q)$ で桁落ちが発生するとTの値も小さくなり、この結果、(42)式の $(T-T')$ の値も小さいとみなされ、結果的に誤差関数に影響するのは、(42)式のC0を含んだ $(T-T')$ がかかっていない項とみなすことができる。このため、桁落ちが発生するときには、むしろ誤差の影響は軽減されると考えられるので、今回は、桁落ちは考慮しないことにする。

【0094】以上述べたように、初期パラメータから誤差関数を計算する際に混入する丸めによる相対誤差は、 $(1+\varepsilon)^{294}$ となる。上で述べた丸め誤差に対する条件から、この相対誤差が、倍精度浮動小数点に含まれる相対誤差以下であることが要請されるため、積和演算器による1回の相対誤差 ε に対して、以下の式が成り立つ。

【0095】

$$(47)$$

仮数部の桁数は、62ビットである。

【0096】前述の文献2のような例では、通常、誤差関数の計算と、2電子積分やフォック行列要素の計算とに用いられる積和演算は、共通の演算装置が使用される。したがって、仮数部が62ビット以上の積和演算

は、誤差関数だけでなく、2電子積分やフォック行列要素を求めるときにも使われる。

【0097】このように、62ビット以上の積和演算で計算することによって、誤差関数は、前述の文献1に目安として示された精度を持つため、分子軌道計算の結果にも十分な精度を持たせることができると考えられる。

【0098】しかしながら、特開平9-50428号公報に示されている方法では、既存のホストとなるベクトル計算機と、計算機クラスタを構成する汎用計算機を用いて分子軌道計算を行っている。このため、ホストとなるベクトル計算機や、クラスタを構成する計算機に予め用意されている演算精度によって、誤差関数の計算や、固有値の演算精度が制限されてしまう問題がある。

【0099】特に、通常の計算機では、IEEE規格754で定められている倍精度浮動小数点表現に対応する乗算回路や加減算回路が用意されているため、(1)式で表される固有値の演算精度は上記の要求を満足するものの、誤差関数の計算には演算精度が不足するものになってしまう。

【0100】ここで、4倍精度浮動小数点などの高精度な演算を用意している計算機を、クラスタとして使用することもできるが、その場合には、誤差関数の計算に十分な演算精度よりもかなりオーバースペックとなってしまうため、ハードウェア資源を無駄に使用してしまうことになり、効率が悪い。

【0101】さらに、IEEE規格754で定められている倍精度浮動小数点と4倍精度浮動小数点のフォーマットの変換に手間がかかる。例えば、倍精度浮動小数点と4倍精度浮動小数点では、指数の値に対して特定のオフセットを加算したものを、指数を表すフィールドに格納するが、倍精度浮動小数点と4倍精度浮動小数点とで、この加算するオフセットの値が異なるため、両者の間で指数部に対する加減算処理を行う等の処理が必要となる。

【0102】また、倍精度浮動小数点に対応したハードウェアを使い、ソフトウェア的に4倍精度浮動小数点などの、より精度の高い計算を行うことも可能であるが、計算に余分なステップ数を要するため、その演算速度が低下してしまう。

【0103】このように、目的とするフォーマットよりも小さい回路を用いて、そのフォーマットの計算を行う例としては、特開平2-171923号公報、特開平6-301710号公報、特開平8-185309号公報に記載のものも挙げられるが、いずれも演算時間がかかってしまう問題がある。

【0104】以上の点にかんがみ、この発明は、オーバースペックになるようなハードウェア規模とする必要がなく、また、計算速度も高速を保ちながら、高精度の並列計算ができるようにした装置および方法を提供することを目的とする。

【0105】

【課題を解決するための手段】上記課題を解決するために、請求項1の発明においては、ホストプロセッサと、このホストプロセッサとバスを介して接続される1または複数のプロセッサエレメントとからなり、前記ホストプロセッサと、前記1または複数のプロセッサエレメントとが、特定の計算プロセスに含まれる浮動小数点演算処理を分担して計算する並列計算装置において、前記プロセッサエレメントは、前記ホストプロセッサとの間で、1ビットの符号部と、 m_e ビットのビット幅の指数部と、 m_0 ビットのビット幅の仮数部とからなる第1の形式の浮動小数点データを入力する浮動小数点入力インタフェース部と、前記浮動小数点入力インタフェース部より入力される前記第1の形式の浮動小数点データを、1ビットの符号部と、前記 m_e ビットのビット幅の指数部と、前記 m_0 よりも大きい m_1 ビットのビット幅の仮数部とからなる第2の形式の浮動小数点データに変換する入力データ変換部と、浮動小数点乗算手段および浮動小数点加減算手段を備え、前記入力データ変換部からの前記第2の形式の浮動小数点データについて、浮動小数点演算処理を実行する浮動小数点演算部と、前記浮動小数点演算部で演算が施された前記第2の形式の浮動小数点データを前記第1の形式の浮動小数点データに変換して、前記浮動小数点入力インタフェース部に供給する出力データ変換部と、を備えることを特徴とする。

【0106】また、請求項2の発明は、請求項1に記載の並列計算装置において、前記入力データ変換部は、前記第1の形式の浮動小数点データの符号部を、前記第2の形式の浮動小数点データの符号部とし、前記第1の形式の浮動小数点データの指数部を、前記第2の形式の浮動小数点データの指数部とし、前記 m_0 ビットの前記第1の形式の浮動小数点データの仮数部を、前記 m_1 ビットの前記第2の形式の浮動小数点データの仮数部のうちの上位の前記 m_0 ビットとし、前記第2の形式の浮動小数点データの仮数部の下位 $m_1 - m_0$ ビットを所定の数値とし、前記出力データ変換部は、前記第2の形式の浮動小数点データの符号部を、前記第1の形式の浮動小数点データの符号部とし、前記第2の形式の浮動小数点データの指数部を、前記第1の形式の浮動小数点データの指数部とし、前記第2の形式の浮動小数点データの仮数部の上位の前記 m_0 ビットを、前記第1の形式の浮動小数点データの仮数部としたことを特徴とする。

【0107】また、請求項3の発明は、請求項1に記載の並列計算装置において、前記入力データ変換部は、前記第1の形式の浮動小数点データの符号部を、前記第2の形式の浮動小数点データの符号部とし、前記第1の形式の浮動小数点データの指数部を、前記第2の形式の浮動小数点データの指数部とし、前記 m_1 ビットの前記第2の形式の浮動小数点データの仮数部のうちの、その最上

位 1 ビットは 1 とし、最上位よりも下位の前記 $m0$ ビットは前記第 1 の形式の浮動小数点データの仮数部とし、さらにそれより下位の前記 $m1$ - 前記 $m0$ - 1 ビットは所定の数値とし、前記出力データ変換部は、前記第 2 の形式の浮動小数点データの符号部を、前記第 1 の形式の浮動小数点データの符号部とし、前記第 2 の形式の浮動小数点データの指数部を、前記第 1 の形式の浮動小数点データの指数部とし、前記第 2 の形式の浮動小数点データの仮数部の上位 2 ビット目からの $m0$ ビットを、前記 $m0$ ビットの第 1 の形式の浮動小数点データの仮数部としたことを特徴とする。

【0108】また、請求項 4 の発明は、請求項 1 ~ 3 のいずれかに記載の並列計算装置において、前記第 2 の形式の浮動小数点データにおいては、前記指数部が第 1 の所定の値 $emax$ のときに符号付き無限大を表し、前記指数部が第 2 の所定の値 $emin$ のときに零を表し、前記浮動小数点演算部に含まれる浮動小数点乗算手段は、オーバーフローを発生すると出力データの指数部を前記値 $emax$ とし、アンダーフローを発生すると出力データの指数部を前記値 $emin$ とし、入力される前記第 2 の形式の 2 つの浮動小数点データのうち、どちらか 1 つの指数部が前記値 $emax$ のときに出力データの指数部を前記値 $emax$ とし、前記入力される前記第 2 の形式の 2 つの浮動小数点データのうち双方とも指数部が前記値 $emax$ でなく、かつ、どちらか 1 つの指数部が前記値 $emin$ のときに出力データの指数部を前記値 $emin$ とし、前記浮動小数点演算部に含まれる浮動小数点加減算手段は、オーバーフローを発生すると出力データの指数部を前記値 $emax$ とし、アンダーフローを発生すると出力データの指数部を前記値 $emin$ とし、入力される前記第 2 の形式の 2 つの浮動小数点データのうち、どちらか 1 つの指数部が前記値 $emax$ のときに出力データの指数部を前記値 $emax$ とすることを特徴とする。

【0109】また、請求項 5 の発明は、請求項 1 ~ 4 のいずれかに記載の並列計算装置において、前記浮動小数点演算部に含まれる浮動小数点乗算手段は、その出力データの仮数部を求めるときに、切り捨てを実施し、前記浮動小数点演算部に含まれる浮動小数点加減算手段は、その出力データの仮数部を求めるときに、切り捨てを実施することを特徴とする。

【0110】また、請求項 6 の発明は、ホストプロセッサと、このホストプロセッサとバスを介して接続される 1 または複数のプロセッサエレメントとにより、特定の計算プロセスに含まれる浮動小数点演算処理を分担して計算する並列計算方法において、前記ホストプロセッサは、1 ビットの符号部と、 me ビットのビット幅の指数部と、 $m0$ ビットのビット幅の仮数部とからなる第 1 の形式の浮動小数点データについて演算処理を行い、前記

で、1 ビットの符号部と、前記 me ビットのビット幅の指数部と、前記 $m0$ よりも大きい $m1$ ビットのビット幅の仮数部とからなる第 2 の形式の浮動小数点データについて、浮動小数点演算処理を実行するものであって、かつ、前記バスを通じて入力される第 1 の形式の浮動小数点データ入力の符号部を前記第 2 の形式の浮動小数点データの符号部とし、前記第 1 の形式の浮動小数点データ入力の指数部を前記第 2 の形式の浮動小数点データの指数部とし、前記第 1 の形式の浮動小数点データ入力の仮数部を、前記 $m1$ ビットの前記第 2 の形式の浮動小数点データの仮数部のうちの上位 $m0$ ビットとすると共に、前記第 2 の形式の浮動小数点データの仮数部の下位 $m1$ - $m0$ ビットを所定の数値として、前記第 1 の形式の浮動小数点データ入力を、前記第 2 の形式の浮動小数点データに変換する入力データ変換工程と、前記入力データ変換工程で変換された前記第 2 の形式の浮動小数点データについて、浮動小数点演算を実行し、その演算結果としての前記第 2 の形式の浮動小数点データを得る浮動小数点演算工程と、前記浮動小数点演算工程で得られる前記演算結果としての前記第 2 の形式の浮動小数点データの符号部を前記第 1 の形式の浮動小数点データの符号部とし、前記演算結果としての前記第 2 の形式の浮動小数点データの指数部を前記第 1 の形式の浮動小数点データの指数部とし、さらに前記演算結果としての前記第 2 の形式の浮動小数点データの仮数部の上位 $m0$ ビットを前記第 1 の形式の浮動小数点データの仮数部として、前記演算結果としての前記第 2 の形式の浮動小数点データを、前記第 1 の形式の浮動小数点データ出力に変換して、前記バスに出力する出力データ変換工程と、を備えることを特徴とする。

【0111】また、請求項 7 の発明は、ホストプロセッサと、このホストプロセッサとバスを介して接続される 1 または複数のプロセッサエレメントとにより、特定の計算プロセスに含まれる浮動小数点演算処理を分担して計算する並列計算方法において、前記ホストプロセッサは、1 ビットの符号部と、 me ビットのビット幅の指数部と、 $m0$ ビットのビット幅の仮数部とからなる第 1 の形式の浮動小数点データについて演算処理を行い、前記プロセッサエレメントは、前記第 1 の形式よりも高精度で、1 ビットの符号部と、前記 me ビットのビット幅の指数部と、前記 $m0$ よりも大きい $m1$ ビットのビット幅の仮数部とからなる第 2 の形式の浮動小数点データについて、浮動小数点演算処理を実行するものであって、かつ、前記バスを通じて入力される第 1 の形式の浮動小数点データ入力の符号部を前記第 2 の形式の浮動小数点データの符号部とし、前記第 1 の形式の浮動小数点データ入力の指数部を前記第 2 の形式の浮動小数点データの指数部とし、さらに $m1$ ビットの第 2 の形式の浮動小数点データの仮数部のうち、その最上位 1 ビットは 1 とし、最上位よりも下位の前記 $m0$ ビットは前記第 1 の形式の

浮動小数点データ入力の仮数部とし、さらにそれより下位の前記 $m1$ - 前記 $m0 - 1$ ビットは所定の数値として、前記第1の形式の浮動小数点データ入力を、前記第2の形式の浮動小数点データに変換する入力データ変換工程と、前記入力データ変換工程で変換されて得られた前記第2の形式の浮動小数点データについて、浮動小数点演算を実行して、その演算結果としての前記第2の形式の浮動小数点データを得る浮動小数点演算工程と、前記浮動小数点演算工程で得られる前記演算結果としての前記第2の形式の浮動小数点データの符号部を前記第1の形式の浮動小数点データの符号部とし、また前記演算結果としての前記第2の形式の浮動小数点データの指数部を前記第1の形式の浮動小数点データの指数部とし、さらに前記演算結果としての前記第2の形式の浮動小数点データの仮数部の上位2ビット目からの $m0$ ビットを、 $m0$ ビットの前記第1の形式の浮動小数点データの仮数部として、前記演算結果としての前記第2の形式の浮動小数点データを、前記第1の形式の浮動小数点データ出力に変換して、前記バスに出力する出力データ変換工程と、を備えることを特徴とする。

【0112】また、請求項8の発明は、請求項1～5のいずれかに記載の並列計算装置において、前記計算プロセスは、分子軌道法に基づく計算プロセスであることを特徴とする。

【0113】さらに、請求項9の発明は、請求項6または請求項7に記載の並列計算方法において、前記計算プロセスは、分子軌道法に基づく計算プロセスであることを特徴とする。

【0114】

【作用】請求項1の発明においては、ホストプロセッサは、第1の形式、例えば倍精度浮動小数点データについて演算処理を実行すると共に、このホストプロセッサと浮動小数点演算処理を分担処理するプロセッサエレメントは、第1の形式の浮動小数点データよりも高精度であって、仮数部のビット幅のみが異なる第2の形式の浮動小数点データを使って浮動小数点演算を行う。したがって、4倍精度を用いる場合のような過剰なオーバースペックとなってしまうようなことはない。

【0115】そして、プロセッサエレメントにおいて、ホストプロセッサから入力データを受けるときに、入力データ変換部により、第1の形式から第2の形式に浮動小数点データの形式の変換を行うと共に、プロセッサエレメントからホストプロセッサに向けてデータを出力するとき、出力データ変換部により、第2の形式から第1の形式に浮動小数点データの形式の変換を行う。

【0116】この浮動小数点データの形式の変換は、仮数部のみが異なるので、前述した4倍精度と倍精度との間での変換のように、指数部の変換を行う必要がない。このため、簡単な変換処理で済むので、演算処理速度の低下が少ない。

【0117】そして、一般に、並列計算装置においては、プロセッサエレメントでの計算量が、ホストプロセッサよりも多いが、上述の構成の請求項1によれば、この計算量の多い演算が、高精度で実施されることになり、精度の高い計算が可能となるものである。

【0118】また、請求項2の発明によれば、入力データ変換部および出力データ変換部で行われる第1の形式の浮動小数点データと第2の形式の浮動小数点データとの間の変換において、符号部と指数部とは、全く同一にすることができる。したがって、変換を単純に行うことができるので、変換に要するハードウェア規模が小さくなる。

【0119】また、請求項3の発明によれば、入力データ変換部および出力データ変換部で行われる第1の形式の浮動小数点データと第2の形式の浮動小数点データとの変換を単純に保ったまま、第2の形式の浮動小数点データの仮数部を浮動小数点乗算手段で計算を行い易いように変更される。したがって、浮動小数点乗算手段の回路規模を小さくできる。

【0120】また、請求項4の発明によれば、第2の形式の浮動小数点データにおいて、零と無限大とが指数部だけで表されるようになる。したがって、浮動小数点乗算手段、および浮動小数点加減算手段で、零や無限大を扱うための回路規模を小さくすることができる。

【0121】また、請求項5の発明においては、浮動小数点乗算手段および浮動小数点加減算手段では、仮数部を求めるときに、丸め処理が省略されて、切り捨て処理が行われる。

【0122】この請求項5の発明の場合、第2の形式の浮動小数点データの仮数部のビット幅が、第1の形式の浮動小数点データの仮数部のビット幅が大きいことから、丸め処理をしても、第1の形式の浮動小数点データとして必要な仮数部のビット幅に影響がないようにすることができる。このため、丸め処理を省略しても、高精度を維持することが可能であるとともに、丸め処理を省略できるので、浮動小数点乗算手段、浮動小数点加減算手段の回路規模を小さくし、高速に演算が実施できるようになる。

【0123】また、請求項6および請求項7の発明によれば、第1の形式の浮動小数点データと第2の形式の浮動小数点データとの間の変換工程によって、汎用的に使用されるプロセッサを使用して変換が高速であり、また、精度の高い浮動小数点演算を実行することができる。

【0124】また、請求項8および請求項9の発明によれば、従来と同程度の速度で、従来よりも高精度で分子軌道計算を実行することができる。

【0125】

【発明の実施の形態】以下、この発明による並列計算装置および並列計算方法の実施の形態を、図を参照して説

明する。

【0126】図1は、実施の形態の並列計算装置のシステム構成のブロック図である。この図1において、1はホストプロセッサであり、既存のパソコンやワークステーションで構成される。2はプロセッサエレメントの例としての専用プロセッサであり、システムに複数個用意される。この複数個の専用プロセッサ2は、所定の個数ごとにまとめられ、そのまとめられた単位はボード3として構成される。ホストプロセッサ1と、複数個のボード3との間は、バス4によって接続される。

【0127】ここで、ボード3上では、バスプロトコル変換用のバスブリッジを介してホストから出るバスとは異なるバスを使用しても良いが、この例では、簡単のために、ホストプロセッサ1から出るバスと、ボード3上のバスとは同じものを使用するものとする。したがって、この例では、バス4は、ホストプロセッサ1と、複数個の専用プロセッサ2との間を接続するものとなっている。

【0128】この図1のシステムは、前述の文献2に記載されているシステムと、ほぼ同じ方法で分子軌道計算を実施する。まず、以下に、このシステムによる分子軌道計算の実施手順を示す。

【0129】(手順1) ボード3上に位置する複数の専用プロセッサ2は、分子軌道計算で必要となる膨大な2電子積分の計算を分担する。このため、まず、ホストプロセッサ1は、各専用プロセッサ2での計算に必要な原子の中心座標A、B、C、Dや、軌道指数 l 、さらに、係数Cの値から、前述の(3)式によって求めた密度行列 P_{tu} などの初期パラメータを、バス4を通して、予め、各専用プロセッサ2に送信する。各専用プロセッサ2では、受け取ったこれらの初期パラメータを、内蔵するメモリに格納する。

【0130】(手順2) 次に、各専用プロセッサ2では、内蔵するメモリに格納された初期パラメータの値を使用して、内蔵されている積和演算回路を使いながら、誤差関数 $F_m(T)$ の計算や、前述の(9)式や(10)式で表される2電子積分の計算、および密度行列の値を使用して、前述の(2)式で表されるフォック行列要素を構成する g_{rs} の計算を実施する。その後、各専用プロセッサ2は、ホストプロセッサ1に対して、求めた g_{rs} の値をバス4を通して送り返す。

【0131】(手順3) ホストプロセッサ1では、前述の(2)式に含まれる h_{rs} を計算し、また、各専用プロセッサ2から送り返された g_{rs} の値とともに、フォック行列要素 F_{rs} を計算する。また、ホストプロセッサ1は、求めたフォック行列要素 F_{rs} の値を元にして、前述の(1)式の固有値問題を解き、新たな係数Cを求める。

【0132】上に述べた手順1～3を繰返すことによって、SCF法が実現される。また、ホストプロセッサ1

は、求められた係数Cの値を、SCFの直前の繰返しで求められた係数の値と比較して、その差が十分に小さければ、SCFの繰返しを終了する。このシステムを使って、以上のような手順で、計算を繰返すことにより、分子のエネルギー値が求まる。

【0133】ここで、この実施の形態では、ホストプロセッサ1で扱う第1の形式の浮動小数点データは、IEEE規格795で定められている倍精度の浮動小数点データとする(以下、単に倍精度という)。すなわち、この実施の形態では、ホストプロセッサ1で実施される浮動小数点演算は、全て倍精度で行われる。また、初期パラメータも倍精度で表されている。この精度は、前述もしたように、ホストプロセッサ1で行われる固有値の演算精度を満足するものである。また、前述の誤差関数の精度に必要な初期パラメータの入力誤差も満足している。

【0134】次に、専用プロセッサ2の構成について説明する。図2は、この実施の形態で用いられている専用プロセッサ2の内部構成を示したブロック図である。この専用プロセッサ2では、後述するように、第1の形式である倍精度の浮動小数点データよりも、仮数部のビット幅が大きい第2の形式の浮動小数点データを扱うものである。

【0135】図2に示すように、専用プロセッサ2は、バスインタフェース回路10と、入力データ変換回路11と、出力データ変換回路12と、積和演算回路20と、メモリ30と、制御回路40と、プログラムメモリ41とを備える。また、内部バス100、101、102と、メモリデータバス103と、メモリアドレスバス104とを備える。

【0136】バスインタフェース回路10は、バス4と専用プロセッサ2とのインタフェースを行う。入力データ変換回路11は、バス4から入力される第1の形式の浮動小数点データを、第2の形式の浮動小数点データに変換する。出力変換回路12は、演算結果をバス4へ出力するために、第2の形式の浮動小数点データを、第1の形式である倍精度の浮動小数点データに変換する。

【0137】積和演算回路20は、第2の形式の浮動小数点データについて積和演算を行うための回路で、第2の形式の浮動小数点データを格納するレジスタ21と、浮動小数点乗算を実行する乗算回路22と、浮動小数点加減算を実行する加減算回路23とを備える。

【0138】メモリ30は、演算前後の第2の形式の浮動小数点データを格納する。制御回路40は、専用プロセッサ2の内部を制御する。プログラムメモリ41は、制御回路40で実行されるプログラムを保持する。

【0139】内部バス100～102は、積和演算回路20に入力するデータを伝達するためのものである。メモリデータバス103は、メモリ30に入出力するデータを伝達する。メモリアドレスバス104は、メモリ3

0 のアドレスを伝達する。なお、200 は、バス 4 と入出力を行うための入出力端子である。

【0140】次に、以上のような構成の専用プロセッサ 2 の機能について説明する。まず、ホストプロセッサ 1 と専用プロセッサ 2 とのデータ入出力機能について述べる。

【0141】専用プロセッサ 2 は、前述のように、第 1 の形式である倍精度の浮動小数点データを、ホストプロセッサ 1 と入出力する。一方、専用プロセッサ 2 の内部では、前述の通り、倍精度よりも高い精度の第 2 の形式の浮動小数点フォーマットで演算を行うことが必要なので、この例では、倍精度の 64 ビットよりも大きい 76 ビットの独自のフォーマット（以下、内部フォーマットと呼ぶ）を用いて計算を行う。内部フォーマットに関する詳しい説明は、後述する。

【0142】このため、ホストプロセッサ 1 から専用プロセッサ 2 にデータを入力するときは、バスインタフェース回路 10 で、バス 4 の伝送手順で入力される信号から倍精度浮動小数点データを構成し、入力データ変換回路 12 によって、倍精度浮動小数点データを 76 ビットの内部フォーマットを持ったデータに変換し、この変換後データをメモリデータバス 103 を経由してメモリ 30 に出力する。

【0143】これと同時に、バスインタフェース回路 10 は、バス 4 から指定されるメモリ 30 のアドレスを生成して、メモリアドレスバス 104 を経由してメモリ 30 に供給する。その後、制御回路 40 は、適切な制御信号を、メモリ 30 に供給することにより、76 ビットデータをメモリ 30 に書き込む。

【0144】また、専用プロセッサ 2 からホストプロセッサ 1 にデータを出力するときは、バスインタフェース回路 10 で、バス 4 から指定されるメモリ 30 のアドレスを生成して、メモリ 30 に供給し、メモリデータバス 103 を経由して、76 ビットの内部フォーマットで表されたデータを読み出す。その後、出力データ変換回路 11 によって、このデータを倍精度浮動小数点データに変換し、バスインタフェース回路 10 で、バス 4 の伝送手順で入力される信号を生成して、バス 4 に出力する。

【0145】次に、積和演算回路 20 の機能について説明する。積和演算回路 20 は、制御回路 40 の制御を受けながら、メモリ 30 やレジスタ 21 に格納された 76 ビットの内部フォーマットで表された浮動小数点データを使用して積和演算を実行し、その結果を、メモリ 30 やレジスタ 21 に格納する。このとき、プログラムメモリ 41 には、積和演算プログラムが予め格納されており、制御回路 40 は、積和演算プログラムを順番に読み出して、それに従った制御信号を生成することにより、積和演算回路 20 やメモリ 30 を制御する。以下に、積和演算回路 20 の各部分の機能をさらに詳しく述べる。

【0146】レジスタ 21 あるいはメモリ 30 からは、

76 ビットの内部フォーマットで表された浮動小数点データが、メモリデータバス 103 を経由して内部バス 100 ~ 102 を通して積和演算回路 20 の内部の乗算回路 22、加減算回路 23 に供給される。

【0147】このとき、レジスタ 21 は、3 ポートの読み出しポートを有しており、各ポートから内部バス 100 ~ 102 に、独立にデータが読み出される。さらに、メモリ 30 から読み出されたデータは、メモリデータバス 103 を通して、内部バス 100 ~ 102 のどれかに入力される。

【0148】読み出しを行うときのレジスタ 21 やメモリ 30 へのアドレス、制御信号は、ともに制御回路 40 で、プログラムメモリ 41 の積和演算プログラムに従って生成される。

【0149】積和演算回路 20 内部の乗算回路 22、加減算回路 23 は、内部バス 100 ~ 102 を通して供給されたデータを使用して積和演算を実行し、その演算結果を出力する。図 2 に示すように、乗算回路 22 は、内部バス 101、102 からデータを入力し、また、加減算回路 23 は、乗算回路 22 の結果と内部バス 100 からデータを入力する。この構成によって積和演算が実現され、加減算回路 23 の出力が、積和演算結果となって出力される。

【0150】また、制御回路 40 は、積和演算プログラムに従って、乗算回路 22、加減算回路 23 の制御信号を生成して、積和演算回路 20 に供給し、積和演算回路 20 の動作タイミングを制御する。

【0151】また、乗算回路 22、加減算回路 23 は、入力を、そのまま出力にバイパスする機能を備えており、この機能は、制御回路 40 からの制御信号によって制御される。加減算回路 23 の、乗算回路 22 からの入力を、出力にバイパスすることにより乗算が実現され、また、乗算回路 22 の片方の入力を出力にバイパスすることによって加算や減算が実現される。積和演算回路 20 の好適な構成として、この例では、パイプライン動作するものが用いられる。

【0152】積和演算回路 20 で求められた演算結果は、加減算回路 23 から出力され、レジスタ 21、または、メモリ 30 に書き込まれる。メモリ 30 には、メモリデータバス 103 を経由して積和演算の結果が供給される。このときのレジスタ 21 やメモリ 30 へのアドレスおよび制御信号は、読み出し時と同様にして、積和演算プログラムに基づいて、制御回路 40 によって生成され、供給される。

【0153】専用プロセッサ 2 は、以上のように動作するので、2 電子積分の計算や、(2) 式の g_{rs} の計算を行うことが可能である。そこで、次に、専用プロセッサ 2 が、以上述べた機能を使って、ホストプロセッサ 1 から初期パラメータを受け取り、(2) 式の g_{rs} を計算するまでの手順について説明する。

【0154】専用プロセッサ2は、まず、軌道指数 g や座標A、B、C、D、さらに密度行列 Ptu や(42)式に示したテーラー展開の係数などを、倍精度浮動小数点データの形式で、ホストプロセッサ1から受け取り、入力データ変換回路11で、76ビットの内部フォーマットに変換した後、メモリ30に格納する。

【0155】次に、予め、プログラムメモリ41に格納された積和演算プログラムの計算手順に従って、メモリ30に格納した初期パラメータを読み出しながら、積和演算回路20を使用して、前述した(1)～(28)式の通りに、誤差関数 $F_m(T)$ 、 $(s * s^*, s * s^*)^{(m)}$ の値、2電子積分の値、 grs の値を順番に計算し、求められた grs の値をメモリ30に書き込む。このときの演算は、全て76ビットの内部フォーマットを使用して行われる。

【0156】また、この計算の途中で現れる中間結果は、レジスタ21やメモリ30に格納して使用する。さらに、(25)式に現れる $(1/2)$ 乗の逆数は、ニュートン法により、(27)式に現れる exp 計算は、テーラー展開により、前述した逆数や誤差関数の計算と同様に、積和演算の繰り返しによって計算される。したがって、全ての計算は、積和演算回路20を使って求められる。

$$v = (-1)^{sign} \times 2^{exp} \times (1. xxx \dots) \quad (48)$$

と正規化し、 $sign$ を符号部 s に、2進数で表された最も右側の $(1. xxx \dots)$ の部分の小数点以下の部分 $xxx \dots$ のうち上位52ビットを仮数部 f_0 に、また、正規化するための指数 exp を指数部 e に、それぞれ格納する。

【0161】但し、 exp の値は負になりうるため、予め、1023のオフセットが加えられて、指数部 e に格納される。このことにより、指数部 e の値を1～2046として、 exp の値は、-1022～+1023の範囲で指定することができる。

【0162】また、指数部 e の値が0のときには、特別な数字を表しており、そのときの仮数部 f_0 が0なら実数 $v=0$ であり、そのときの仮数部 f_0 が0でないときには、実数 v は、(48)式に示した正規化ができないほど、その絶対値が小さい数を表すものとされる(非正規化数と呼ぶ)。

【0163】また、指数部 e の値が2047のときにも特別な数字を表しており、そのときの仮数部 f_0 が0なら実数 $v=\pm\infty$ であり、そのときの仮数部 f_0 が0でなければ、実数 v は、例えば $0 \times \infty$ の計算を行った結果のように、不定となった数を表すものとされる。

【0164】次に、この実施の形態の専用プロセッサ2の内部フォーマットの浮動小数点表現について説明する。

【0165】図3(B)に示したように、この内部フォーマットの浮動小数点データは、1ビットの符号部 s

【0157】最後に、このようにしてメモリ30に格納された内部フォーマットの grs の値を、出力データ変換回路12で、倍精度浮動小数点データに変換した後、バスインタフェース回路10によってバス4に出力し、ホストプロセッサ1に送る。以上のような計算手順によって、上述の実施の形態のシステムを用いた分子軌道計算が実現される。

【0158】次に、この実施の形態で使用する、76ビット浮動小数点の内部フォーマットについて説明する。図3(A)に、IEEE規格754で決められている倍精度浮動小数点のフォーマットを、また、図3(B)に、この実施の形態で使用する専用プロセッサ2の内部フォーマットを、それぞれ示す。

【0159】まず、IEEE規格754で定められている倍精度浮動小数点フォーマットについて説明する。図3(A)に示したこのフォーマットは、1ビットの符号部 s と、11ビットの指数部 e と、52ビットの仮数部 f_0 とからなる。このフォーマットによって表現される実数 v は、図3(A)に示した式に基づいて計算される。

【0160】すなわち、この倍精度浮動小数点のフォーマットで、0以外の実数 v を、浮動小数点で表現するときには、

と、11ビットの指数部 e と、64ビットの仮数部 f_1 とからなる。1ビットの符号部 s と、11ビットの指数部 e は、上述した倍精度浮動小数点フォーマットの符号部 s と、指数部 e と同じ構成である。ただし、仮数部 f_1 は64ビットとされて、倍精度浮動小数点フォーマットよりも高精度の浮動小数点フォーマットとしている。

【0166】この内部フォーマットの指数部 e には、 $e=0$ 、 $e=2047$ の場合を除き、倍精度表現と同じ意味の値が格納される。さらに、仮数部 f_1 は、実数 v を、前述した(48)式のように正規化したときに、 $1. xxx \dots$ のうちの $xxx \dots$ の部分の上位64ビットを格納したものとなる。

【0167】この内部フォーマットは倍精度浮動小数点と異なり、非正規化数、不定の数に対する表現は持たない。したがって、仮数部 f_1 の値が0であるか否かに拘らず、指数部 $e=0$ のときには、実数 $v=0$ であり、また、指数部 $e=2047$ のときには、実数 $v=\pm\infty$ を表しているものとする。

【0168】この内部フォーマットのように64ビットの仮数部を持たせると、小数点の上位にある1も含めて65ビットの精度となるため、前述のように誤差関数の計算に必要な62ビットの仮数部の精度を満足している。この場合に、本来ならば、仮数部の精度は、62ビット丁度あれば良いが、この実施の形態では、仮数部の長さを2の冪乗の数とすることによって、回路が構成しやすい場合があるため、3ビット余裕を持たせて64

ビットとしている。

【0169】次に、この実施の形態の専用プロセッサ2の内部フォーマットを決定した背景について述べる。

【0170】特に、この実施の形態の専用プロセッサの内部フォーマットは、仮数部が64ビットにも拘わらず、従来の4倍精度フォーマットなどとは異なり、指数部eの長さが、IEEE754規格の倍精度フォーマットと同じになっている。

【0171】この実施の形態のように、特定のアルゴリズムに従って動作する専用計算機のプロセッサは、ある程度、決められた範囲の値を入力して、特定の手順で計算を行うので、その計算の途中結果が取り得る範囲は限定されている。このため、アルゴリズムの工夫などによって、内部表現の指数部は、倍精度の浮動小数点表現と同様の11ビットで表現することが可能である。このことにより、内部フォーマットとしての第2の形式の浮動小数点データを短くでき、このデータを格納するメモリ容量を減少させることができるとともに、前述したオフセットの値も同じ1023にできるため、指数を計算する回路規模も減少させることができる。

【0172】また、この内部フォーマットを使用した場合、ホストプロセッサ1から専用プロセッサ2に送付される初期パラメータに、万が一、 ∞ や不定が混入していた場合には、エラーを出して計算自体を無効にするなどの措置をホストプロセッサ1で講じれば良いので、専用プロセッサ2に対して、 ∞ や不定は送付されないとみなすことができる。

【0173】また、初期パラメータに、万が一、非正規化数が出てくる場合にも、ホストプロセッサ1で、予め、値に定数をかけて、非正規化数でないようにすれば良い。あるいは専用プロセッサ2で、0とみなしても良い場合には、非正規化数でなく、0の値として送付すれば良い。

【0174】さらに、きわめてまれに、専用プロセッサ2での計算の途中結果が、11ビットの指数部eで表せないような非常に小さい数になる場合、あるいは非常に大きい数になる場合や、乗算器で $0 \times \infty$ などの計算が発生することもあり得る。このような場合は、積和演算回路20による計算結果の指数部eが、0か、あるいは2047かを判定することで検出できる。

【0175】したがって、専用プロセッサ2は、指数部eが0か、あるいは2047かを判定することにより、このような場合を検出したら、非正規化数や不定をホストプロセッサ1に送付することなく、専用プロセッサ2からホストプロセッサ1に割り込みを発生したり、特定のフラグを立てるなどして通知するようにすることができる。この通知を受けたホストプロセッサ1では、エラーを発生させたり、計算結果を0にするなど、適切に処理することができる。

【0176】以上のようにして、この実施の形態の76

ビットの内部フォーマットでは、非正規化数や不定に対する表現は持たず、非常に小さい数は0、非常に大きい数や不定の数は ∞ とみなすことで不都合を生じないようにしている。

【0177】このように、IEEE規格754では、指数部eと仮数部f0の両方によって、0や非正規化数、あるいは ∞ や不定を表しているが、この実施の形態の専用プロセッサ2の内部フォーマットの場合には、非正規化数や不定に対する表現は不要であり、この結果、内部フォーマットでは指数部eの値が、0あるいは2047のときには、仮数部f1の値に関係なく、0と ∞ を表すようにしている。したがって、積和演算回路20の乗算回路22や加減算回路23中の例外的な処理を実施するための回路が単純になる。

【0178】次に、この実施の形態において、76ビット内部フォーマットを用いた乗算、加減算の方法について説明する。

【0179】まず、図4を参照して、乗算回路22の構成と動作について説明する。この実施の形態で使用される乗算回路22は、図4のブロック図に示すように、乗算コア回路300と、オール0判定回路301、302と、オール1判定回路303、304と、オアゲート310、311、312と、マルチプレクサ313、314、315とからなる。

【0180】乗算コア回路300は、76ビットの内部フォーマットで表された入力Aと、入力Bの浮動小数点データどうしの乗算を行う。

【0181】オール0判定回路301および302は、76ビットの内部フォーマットで表現された入力Aおよび入力Bの11ビットの指数部を、それぞれ入力として受けて、その値が全て0のときだけ、出力を“1”にする。また、オール1判定回路303および304は、入力Aおよび入力Bの11ビットの指数部を、それぞれ入力として受けて、その値が全て1のときだけ、出力を“1”にする。

【0182】オアゲート310は、乗算コア回路300のアンダーフロー出力ufと、オール0判定回路301、302の出力の論理和をとる。オアゲート311は、乗算コア回路300のオーバーフロー出力ofと、オール1判定回路303、304の出力の論理和をとる。さらに、オアゲート312は、オアゲート310と311の論理和をとる。

【0183】マルチプレクサ313は、オアゲート310および311の出力を2ビットの選択制御信号SE1として受けて、オール0と、オール1と、乗算コア回路300の乗算出力M1のうちの指数部データとの3通りの11ビットの指数部データのうちの1つを選び出す。

【0184】マルチプレクサ314は、オアゲート312の出力を選択制御信号SE2として受けて、オール0と、乗算コア回路300の乗算出力M1のうちの仮数部

10

20

30

40

50

データとの2通りの64ビットの仮数部のうちの1つを選び出す。

【0185】また、マルチプレクサ315は、乗算回路22の出力MOとする76ビットの内部フォーマットで表現された浮動小数点データを、2通りのデータから選択する。選択される2通りのデータの一方は、乗算回路22をバイパスするときの入力Aである。他方は、乗算コア回路300の乗算出力M1のうちの符号部と、マルチプレクサ313で選択された指数部と、マルチプレクサ314で選択された仮数部からなる76ビットの内部フォーマットで表現された最終乗算結果M2である。

【0186】マルチプレクサ315は、制御回路40から入力する選択制御信号によって、出力MOとして最終乗算結果M2を出力するか、バイパス入力データAを出力するかを選択する。

【0187】次に、以上のような構成の乗算回路22の動作について、説明する。まず、内部フォーマットで表された2つの入力Aと入力Bとは、乗算コア回路300によって乗算が行われ、内部フォーマットで表された乗算結果M1が、乗算コア回路300から出力される。

【0188】この際、乗算コア回路300においては、2入力A、Bの仮数部どうしの乗算結果は、丸められて乗算結果M1の仮数部が得られる。この丸めにより、65ビットの演算精度が得られる。また、乗算コア回路300から出力される内部フォーマットの乗算結果M1の指数部と仮数部とは、乗算コア回路300の入力および出力の指数部eフィールドが、1~2046の範囲にある場合だけに対して正しい結果となっていれば良い。

【0189】ここで、図3に示したように、内部フォーマットは、倍精度浮動小数点の仮数部のビット幅を拡張しただけなので、このような乗算コア回路は、従来公知の倍精度浮動小数点に用いる乗算回路を用いて、その仮数部のビット幅を単純に拡張するだけで構成できる。したがって、この実施の形態における乗算コア回路300の内部構成の詳細は省略する。

【0190】また、乗算コア回路300では、 (1×2^{-1000}) どうしの乗算のように、入力Aと入力Bとは内部フォーマットで表されているにも拘わらず、指数部が11ビットの範囲を超えてしまうために、乗算出力が内部フォーマットで表せないほど小さくなる（アンダーフローする）場合には、アンダーフロー出力ufから

“1”が出力される。

【0191】また、 $(1 \times 2^{+1000})$ どうしの乗算のように、入力Aと入力Bとは内部フォーマットで表されるにも拘わらず、指数部が11ビットの範囲を超えてしまうために、乗算出力が内部フォーマットで表せないほど大きくなる（オーバーフローする）場合には、オーバーフロー出力ofから“1”が出力される。

【0192】さらに、内部フォーマットの入力A、入力Bの指数部は、オール0判定回路301、302および

オール1判定回路303、304に入力されているため、オール0判定回路301からは、入力Aが0のとき、“1”が出力され、また、オール0判定回路302からは、入力Bが0のとき、“1”が出力され、また、オール1判定回路303からは、入力Aが ∞ のとき、“1”が出力され、また、オール1判定回路304からは、入力Bが ∞ のとき、“1”が出力される。

【0193】次に、乗算コア回路300によって生成された内部フォーマットの乗算結果M1から、内部フォーマットの最終乗算結果M2を生成する方法について述べる。

【0194】まず、乗算結果M1の符号部は、そのまま、最終乗算結果M2の符号部となる。

【0195】次に、最終乗算結果M2の指数部は、マルチプレクサ313によって出力される。このとき、オアゲート310には、乗算コア回路300のアンダーフロー出力ufと、オール0判定回路301、302の判定結果とが入力されているため、乗算がアンダーフローした場合、あるいは入力Aか入力Bが0の場合に、オアゲート310の出力は“1”になる。また、オアゲート311には、乗算コア回路300のオーバーフロー出力ofと、オール1判定回路303、304の判定結果とが入力されているため、乗算がオーバーフローした場合、あるいは入力Aか入力Bが ∞ の場合に“1”になる。

【0196】これらのオアゲート310、311の出力は、2ビットの選択制御信号SE1としてまとめられ、マルチプレクサ313に入力されている。そして、マルチプレクサ313からは、選択制御信号SE1が“00”のときには乗算結果M1の指数部が出力され、“10”のときには11ビットの0（オール0）が出力され、それ以外のときには11ビットの1（オール1）が出力される。

【0197】さらに、最終乗算結果M2の仮数部としては、マルチプレクサ314から、オアゲート312からの選択制御信号SE2に応じ、信号SE2が“0”のときには乗算結果M1の仮数部が、“1”のときには64ビットの0（オール0）が出力される。オアゲート312は、オアゲート310、311の論理和出力を選択制御信号SE2として出力するので、乗算結果がオーバーフローした場合、またはアンダーフローした場合、または、入力A、入力Bのうち、0か ∞ のものが1つでもある場合に、選択制御信号SE2が“1”になり、マルチプレクサ314からは最終乗算結果M2の仮数部として64ビットの0が出力され、その他のときには、乗算結果M1の仮数部が出力される。

【0198】以上の結果、乗算結果M1がオーバーフローもアンダーフローもせず、また、入力Aも入力Bも0や ∞ でない場合には、選択制御信号SE1は“00”、選択制御信号SE2は“0”になるので、最終乗算結果M2には乗算結果M1が、そのまま出力される。

【0199】また、乗算結果がアンダーフローするか、または、入力Aか入力Bが、0の場合で、かつ、入力A、Bのどちらも ∞ でない場合には、選択制御信号SE1が“10”、選択制御信号SE2が“1”になるので、最終乗算結果M2の指数部は0になって、内部フォーマットの0になる。なお、最終乗算結果M2の仮数部も、マルチプレクサ314によって0になるので、後述するように、内部フォーマットの64ビット仮数部から上位52ビットを抽出することにより、IEEE規格754の倍精度浮動小数点数を取り出しても、0を表す表現となっている。

【0200】また、乗算結果がオーバーフローするか、または、入力Aか入力Bが ∞ の場合で、かつ、入力A、Bのどちらも0でない場合には、選択制御信号SE1が“01”、選択制御信号SE2が“1”になるので、最終乗算結果M2の指数部は、2047となり、内部フォーマットの ∞ になる。なお、最終乗算結果M2の仮数部も、マルチプレクサ314によって0になるので、後述するように、内部フォーマットの64ビット仮数部から上位52ビットを抽出してIEEE規格754の倍精度浮動小数点数を取り出しても、 ∞ を表す表現となっている。

【0201】さらに、入力Aと入力Bの一方が0、他方が ∞ の場合には、選択制御信号SE1が“11”、選択制御信号SE2が“1”になるため、最終乗算結果M2は、上述の ∞ を出力するときと同じ出力になる。この場合は、結果を不定にするべきところで、IEEE規格754で ∞ を表す表現となっている。しかし、この実施の形態のようなシステムでは、 ∞ と不定とを区別する必要がないので、このまま出力すれば良い。

【0202】最後に、最終乗算結果M2は、マルチプレクサ315によって、入力Aのバイパスデータとの間で選択される一方の出力MOとして出力される。ここで、入力Aのバイパス路は、前述のように、加減算命令を実行する場合などのように、バイパスを行うために設けられている。

【0203】なお、前述の選択制御信号SE1は、フラグ信号FLAG1としても出力され、図2に示した破線の経路を通じて制御回路40に入力される。制御回路40は、このフラグ信号FLAG1による割り込みなどによって、乗算結果が0や ∞ になったことを検知して、それに対応した適当な処理プログラムを実行することができる。

【0204】以上のように、上に述べた乗算回路は、公知の倍精度浮動小数点乗算回路に対してほぼ必要な精度を満足するビット数だけを拡張した構成をとるので、回路規模の増加を最小限に抑えることができ、かつ必要な精度の計算を高速に実施することができる。

【0205】次に、図5を参照しながら加減算回路23の構成と動作について説明する。この実施の形態で使用

される加減算回路23は、図5のブロック図に示すように、加減算コア回路400と、オール1判定回路401、402と、オアゲート411、412と、マルチプレクサ413、414、415とからなる。

【0206】加減算コア回路400は、76ビットの内部フォーマットで表された入力Aと、入力Bの浮動小数点データどうしの加減算を行う。オール1判定回路401、402は、76ビットの内部フォーマットで表された入力Aおよび入力Bの11ビットの指数部を、それぞれ入力として受けて、その値が全て1のときだけ、出力を“1”にする。

【0207】オアゲート411は、加減算コア回路400のオーバーフロー出力ofと、オール1判定回路401、402の出力の論理和をとる。オアゲート412は、加減算コア回路400のアンダーフロー出力ufと、オアゲート411の出力との論理和をとる。

【0208】マルチプレクサ413は、加減算コア回路400のアンダーフロー出力ufと、オアゲート411の出力との2ビットを、選択制御信号SE3として受けて、オール0と、オール1と、加減算コア回路400の加減算出力D1のうちの指数部データとの3通りの11ビットの指数部データのうちの1つを選び出す。

【0209】マルチプレクサ414は、オアゲート412の出力を選択制御信号SE4として受けて、オール0と、加減算コア回路400の加減算出力D1のうちの仮数部データとの2通りの64ビットの仮数部のうちの1つを選び出す。

【0210】また、マルチプレクサ415は、加減算回路23の出力DOとする76ビットの内部フォーマットで表現された浮動小数点データを、2通りのデータから選択する回路である。選択される2通りのデータの一方は、加減算回路23をバイパスするときの入力Aである。他方は、加減算コア回路400の加減算出力D1のうちの符号部と、マルチプレクサ413で選択された指数部と、マルチプレクサ414で選択された仮数部からなる76ビットの内部フォーマットで表現された最終加減算結果D2である。

【0211】マルチプレクサ415は、制御回路40から入力する選択制御信号によって、出力DOとして、最終加減算結果D2を出力するか、バイパス入力データAを出力するかを選択する。

【0212】次に、以上のような構成の加減算回路23の動作について、説明する。まず、内部フォーマットで表された2つの入力Aと入力Bとは、加減算コア回路400によって加減算が行われ、内部フォーマットの加減算結果D1が、加減算コア回路400から出力される。

【0213】この際、加減算コア回路400においては、2入力A、Bの仮数部どうしの加減算結果は、丸められて、加減算結果D1の仮数部が得られる。この丸めにより、65ビットの演算精度が得られる。また、加減

算コア回路 400 から出力される内部フォーマットの加減算結果 D1 の指数部と仮数部とは、加減算コア回路 400 の出力の指数部 e フィールドが、1~2046 の範囲にある場合だけに対して正しい結果となっていれば良い。

【0214】ここで、図 3 に示したように、内部フォーマットは、倍精度浮動小数点データの仮数部のビット幅を拡張しただけなので、このような加減算コア回路 400 は、従来公知の倍精度浮動小数点データに対する加減算回路を用いて、その仮数部のビット幅を単純に拡張するだけで構成できる。したがって、この実施の形態における加減算コア回路 400 の内部構成の詳細は省略する。

【0215】ここで、加減算回路 23 は、乗算回路 22 とは異なり、入力的一方が 0 の場合でも、入力の他方が適当な範囲の値を持っていれば、出力は内部フォーマットで表せる範囲の値となるため、乗算回路 22 のように、入力が 0 の場合に対応した例外的な処理は設けない。

【0216】また、加減算コア回路 400 では、0 どちらの加算や、 $(1 \times 2^{+2}) - (1 \times 2^{+2})$ の計算など、計算結果が内部表現で 1 以上の指数部 e の値で表せないほど小さくなるときには、アンダーフローしたとみなしてアンダーフロー出力 u f から “1” が出力される。

【0217】さらに、 $(1.1 \times 2^{+1023})$ どちらの加算のように、入力 A と入力 B とは、内部フォーマットで表されるにも拘わらず、指数部が 11 ビットの範囲を超えてしまうために、出力が内部フォーマットで表せないほど大きくなる（オーバーフローする）場合には、オーバーフロー出力 o f から “1” が出力される。

【0218】また、入力 A の指数部は、オール 1 判定回路 401 に入力され、入力 B の指数部は、オール 1 判定回路 402 に入力されているため、それぞれの判定回路 401、402 からは、入力 A、入力 B が、 ∞ であるときに “1” が出力される。

【0219】次に、加減算コア回路 400 から出力された加減算結果 D1 から、最終加減算結果 D2 を生成する方法について説明する。

【0220】まず、最終加減算結果 D2 の符号部には、加減算結果 D1 の符号部がそのまま出力される。

【0221】次に、最終加減算結果 D2 の指数部は、マルチプレクサ 413 により選択される。このマルチプレクサ 413 の選択制御信号 S E 3 を生成するオアゲート 411 には、加減算コア回路 400 のオーバーフロー出力 o f と、オール 1 判定回路 401、402 の判定結果とが入力されているため、加減算がオーバーフローした場合、あるいは入力 A か入力 B が ∞ の場合に、オアゲート 411 の出力が “1” になる。

【0222】オアゲート 411 の出力は、加減算コア回路 400 のアンダーフロー出力 u f とともに 2 ビットの

制御信号 S E 3 とされ、マルチプレクサ 413 を制御する。マルチプレクサ 413 は、2 ビットの選択制御信号 S E 3 が “00” のときは、加減算結果 D1 の指数部を出力し、“10” のときには 11 ビットの 0（オール 0）を出力し、また、“01” あるいは “11” のときには 11 ビットの 1（オール 1）を出力する。

【0223】さらに、最終加減算結果 D2 の仮数部は、マルチプレクサ 414 から出力される。このマルチプレクサ 414 の選択制御信号 S E 4 を出力するオアゲート 412 は、加減算コア回路 400 のアンダーフロー出力 u f と、オアゲート 411 の出力との論理和をとることで、加減算結果 D1 がオーバーフローした場合や、アンダーフローした場合、または、入力 A、B のうち ∞ のものが 1 つでもある場合に、その出力である選択制御信号 S E 4 が “1” になる。

【0224】そして、選択制御信号 S E 4 が、“1” なら 64 ビットの 0（オール 0）が、“0” なら加減算結果 D1 の仮数部が、それぞれマルチプレクサ 414 から出力される。

【0225】以上の結果、加減算結果 D1 が、オーバーフローもアンダーフローもせず、また、入力 A も入力 B も、 ∞ でない場合には、選択制御信号 S E 3 は “00”、制選択御信号 S E 4 は “0” となるため、最終加減算結果 D2 としては、加減算結果 D1 が、そのまま出力される。

【0226】また、加減算結果 D1 がアンダーフローする場合には、選択制御信号 S E 3 が “10” になるので、最終加減算結果 D2 の指数部は、オール 0 になって、内部フォーマットの 0 になる。なお、このとき、選択制御信号 S E 4 は “1” になるので、最終加減算結果 D2 の仮数部もオール 0 になり、後述するように、内部フォーマットの 64 ビット仮数部から、その上位 52 ビットを抽出して、I E E E 規格 754 表現の倍精度浮動小数点数を取り出しても、0 を表す表現となっている。

【0227】さらに、加減算結果 D1 が、オーバーフローするか、または、入力 A か入力 B のうち的一方だけが、 ∞ の場合には、選択制御信号 S E 3 が “01” になるので、最終加減算結果 D2 の指数部は 2047 になり、内部フォーマットの ∞ になる。なお、このとき、選択制御信号 S E 4 は “1” となって、最終加減算結果 D2 の仮数部も 0 になり、後述するように、内部フォーマットの 64 ビット仮数部から、その上位 52 ビットを抽出して倍精度浮動小数点数を取り出しても、I E E E 規格 754 で ∞ を表す表現となっている。

【0228】また、 $\infty - \infty$ の計算を実施するときは、選択制御信号 S E 3 の値が “01” または “11” となる。まず、両入力 A、B の仮数部が等しくなければ、加減算コア回路 400 のアンダーフロー出力は 0 であるので、上記と同様、選択制御信号 S E 3 は “01” になる。

10

20

30

40

50

【0229】次に、 $\infty - \infty$ の計算において、両入力A、Bの仮数部が等しければ、加減算コア回路400の内部での仮数部どうしの減算結果が丁度0になるため、アンダーフロー出力が1になり、この結果、選択制御信号SE3は“11”になる。この場合でも、マルチプレクサ413からは、最終加減算結果D2の指数部として11ビットの1（オール1）が出力されるので、加減算結果D1がオーバーフローする場合と同じく、最終加減算結果D2は、仮数部が0であるため、IEEE754規格で ∞ を表す数字になっている。この場合も、本来は結果を不定とすべきところであるが、この実施の形態のようなシステムでは、 ∞ と不定とを区別する必要がないので、このまま出力すれば良い。

【0230】最後に、最終加減算結果D2は、マルチプレクサ415によって、入力Aのバイパスデータとの間で選択される一方の出力DOとして出力される。ここで、入力Aのバイパス路は、前述のように、乗算命令を実行する場合などのように、バイパスを行うために設けられている。

【0231】また、2ビットの選択制御信号SE3は、フラグ信号FLAG2としても出力され、図2に示した破線の経路を通じて制御回路40に入力される。制御回路40は、このフラグ信号FLAG2によって、割り込みなどにより、加減算結果が0や ∞ になったことを検知し、それに対応した適当な処理プログラムを実行することができる。

【0232】以上のように、上に述べた加減算回路23は、公知の倍精度浮動小数点加減算回路に対してほぼ必要な精度を満足するビット数だけを拡張した構成をとるので、回路規模の増加を最小限に抑えることができ、かつ必要な精度の計算を高速に実施することができる。

【0233】次に、入力データ変換回路11、出力データ変換回路12について説明する。これらの変換回路では、図3（B）に示した内部フォーマットと、図3

（A）に示した倍精度浮動小数点のフォーマットとの変換をする。まず、倍精度浮動小数点フォーマットで表された数を、内部フォーマットに変換するための入力データ変換回路11は、以下の（a）、（b）のようにして、フォーマット変換を行う。

【0234】（a）倍精度浮動小数点フォーマットの符号部と指数部は、そのまま内部フォーマットの符号部と指数部として出力する。

【0235】（b）内部フォーマットの64ビットの仮数部の上位52ビットに、倍精度フォーマットの52ビットの仮数部を出力し、また、内部フォーマットの64ビットの仮数部の下位12ビットに、全て0データを出力する。

【0236】また、出力データ変換回路12は、以下の（c）、（d）のようにして、フォーマット変換を行う。

【0237】（c）内部フォーマットの符号部と指数部とは、そのまま倍精度フォーマットの符号部と指数部に出力する。

【0238】（d）内部フォーマットの64ビットの仮数部は、その上位から52ビットを取り出して、倍精度フォーマットの仮数部に出力する。

【0239】このように、前記（d）のように、出力データ変換回路12では、内部フォーマットの仮数部の53ビット目を、丸めることなしに出力するため、出力データgrs自体の精度は、丸めがある場合は53ビットであるのに対し、この場合は52ビットになるが、grsを基にして（2）式によって計算されるフォック行列要素Frsの精度は、52ビットで十分であると考えられる。

【0240】以上説明したように、この実施の形態では、倍精度浮動小数点よりも高精度の専用プロセッサ2での内部フォーマットとして、図3に示したような、倍精度浮動小数点データとは、仮数部のビット幅が異なり、かつ、その仮数部のビット幅が倍精度浮動小数点の仮数部よりも大きくしたフォーマットを採用したので、簡単な倍精度浮動小数点数との間の変換回路を設けるだけで、専用プロセッサ2による高精度な浮動小数点演算を実現できる。

【0241】なお、上記の実施の形態では、乗算コア回路300、加減算コア回路400では、丸めが行われるため、65ビットの演算精度が確保されていたが、仮数部どうしの乗算結果、あるいは仮数部どうしの加減算結果に対して、丸めを行わず、単純に下位のビットを切り捨てて、結果の仮数部を構成しても良い。

【0242】このように構成した場合、演算精度は丸めを行う場合より1ビット少ない64ビットになるが、誤差関数に必要な演算の精度である62ビットは満足する。また、このような構成とすることにより、丸め回路が不要になるため、回路規模を小さくできるとともに、計算を速くすることができる。

【0243】また、図3に示した内部フォーマットの例では、実数vを、前述の（48）式のように表したときの $x \times x \dots$ の部分の仮数部としていたが、図6に示すように、（48）式の $1 \times x \times x \dots$ の部分の仮数部としても良い。

【0244】このことによって、乗算コア回路300、加減算コア回路400で丸めを行わない場合には、演算精度は、上述の場合よりもさらに1ビット少ない63ビットになるが、誤差関数に必要な演算の精度62ビットは満足する。また、倍精度浮動小数点に対する乗算を行う回路では、仮数部の最上位に1を付加することが必要であったが、この構成にすることにより、乗算コア回路300では仮数部をそのまま乗算するようにすれば良く、回路が簡単になる。

【0245】このようにする場合には、専用プロセッサ

2の入力データ変換回路11では、(e)倍精度の浮動小数点データの符号部を、内部フォーマットの浮動小数点データの符号部とし、(f)倍精度の浮動小数点データの指数部を、内部フォーマットの浮動小数点データの指数部とし、(g)内部フォーマットの浮動小数点データの仮数部のうち、その最上位1ビットは1とし、最上位よりも下位の52ビットは倍精度の浮動小数点データの仮数部とし、さらにそれより下位のビットは0データとすることにより、倍精度の浮動小数点データを、専用プロセッサ2の内部のフォーマットの浮動小数点データに変換する。

【0246】したがって、この例の場合には、内部フォーマットの浮動小数点データの仮数部には、最上位に1が含まれているため、乗算を行う場合に、仮数部の最上位に1を付加することが不要となり、乗算コア回路300では仮数部をそのまま乗算するようにすれば良くなって、回路が簡単になるものである。

【0247】そして、この例の場合、この内部フォーマットの浮動小数点データを、倍精度の浮動小数点データに変換する出力データ変換回路12では、(h)内部フォーマットの浮動小数点データの符号部を、倍精度の浮動小数点データの符号部とし、(i)内部フォーマットの浮動小数点データの指数部を、倍精度の浮動小数点データの指数部とし、(j)内部フォーマットの浮動小数点データの仮数部の上位2ビット目からの52ビットを、倍精度の浮動小数点データの52ビットの仮数部とすることにより、専用プロセッサ2の内部フォーマットの浮動小数点データを、倍精度の浮動小数点データに変換する。

【0248】なお、上述の実施の形態の説明では、プロセッサエレメントとして、専用プロセッサを用意して、倍精度浮動小数点と内部フォーマットとの間の変換や、内部フォーマットでの浮動小数点演算をハードウェアで実施する場合について述べたが、従来の倍精度浮動小数点演算機能が用意されているプロセッサエレメントを使用して、ソフトウェアでフォーマット間の変換や、内部フォーマットでの計算を実施することもできる。

【0249】この場合は、専用プロセッサを用意する場合よりも変換と演算の速度は低下するが、4倍精度を使用する場合よりは、変換などが簡単にできるために高速であり、この発明は、このような場合にも有効である。

【0250】また、上記説明では分子軌道計算を例にとったが、この発明は分子軌道計算以外の計算を行う、ホストプロセッサと、複数のプロセッサエレメントとで浮動小数点による計算を分担するタイプの並列計算装置のいずれにも適用することができることは言うまでもない。

【0251】特に、専用プロセッサが複数個あるような場合には、一般に、演算回数が多い計算が専用プロセッサに割り当てられるため、専用プロセッサでの計算も高

い精度が必要となる。したがって、このような並列計算装置に対しても、この発明は有効である。

【0252】

【発明の効果】以上説明したように、この発明によれば、オーバースペックになるようなハードウェア規模とする必要がなく、また、計算速度も高速を保ちながら、高精度の並列計算ができる。特に、フォーマット変換に必要な回路規模を小さく抑えることができる。

【図面の簡単な説明】

10 【図1】この発明による並列計算装置の実施の形態を示すブロック図である。

【図2】実施の形態のプロセッサエレメントの内部構成を示すブロック図である。

【図3】第1の形式の浮動小数点データのフォーマットの例と、実施の形態で使用される第2の形式の浮動小数点データフォーマットの例を示す図である。

【図4】実施の形態のプロセッサエレメントの積和演算回路内の乗算回路の内部構成の例を示すブロック図である。

20 【図5】実施の形態のプロセッサエレメントの積和演算回路内の加減算回路の内部構成の例を示すブロック図である。

【図6】この発明の別の実施の形態で使用する第2の形式の浮動小数点データフォーマットを示す図である。

【図7】誤差関数の計算において、その経路で発生する丸め誤差を説明するための図である。

【符号の説明】

- 1 ホストプロセッサ
- 2 専用プロセッサ
- 30 3 ボード
- 4 バス
- 10 バスインタフェース回路
- 11 入力データ変換回路
- 12 出力データ変換回路
- 20 積和演算回路
- 21 レジスタ
- 22 乗算回路
- 23 加減算回路
- 30 メモリ
- 40 40 制御回路
- 41 プログラムメモリ
- 100~102 内部バス
- 103 メモリデータバス
- 104 メモリアドレスバス
- 200 入出力端子
- 300 乗算コア回路
- 301、302 オール0判定回路
- 303、304 オール1判定回路
- 310~312 オアゲート
- 50 313~315 マルチプレクサ

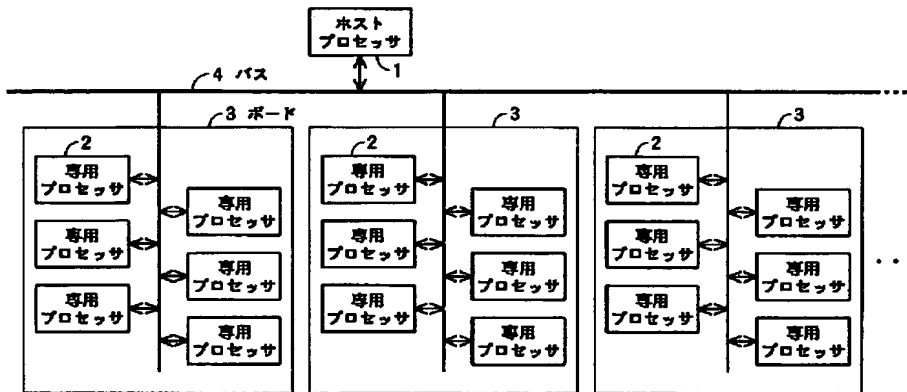
400 加減算コア回路

401, 402 オール1判定回路

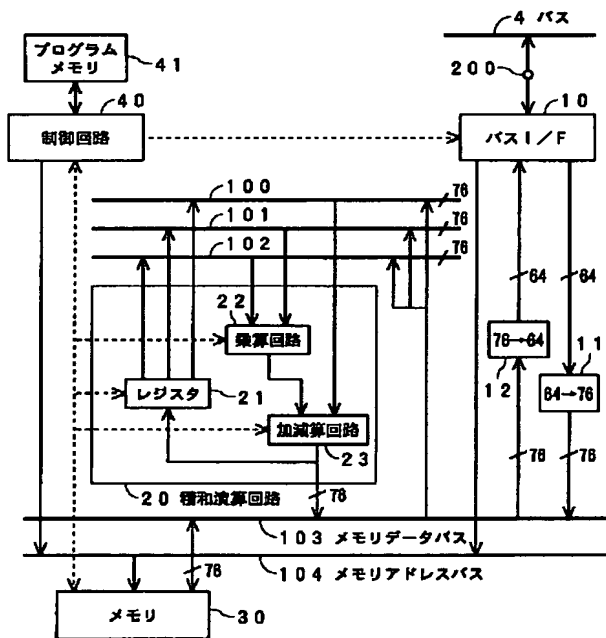
411, 412 オアゲート

413~415 マルチプレクサ

【図1】

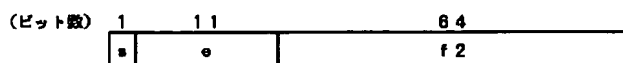


【図2】



【図6】

専用プロセッサ2内での76ビット内部フォーマット表現



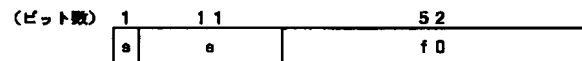
$$v = (-1)^s \times 2^{e-1022} \times (f \times 2^{-64}) \quad (0 < e < 2047)$$

$$= (-1)^s \times \infty \quad (e = 2047)$$

$$= 0 \quad (e = 0)$$

【図3】

(A) IEEE規格754の倍精度表現



$$v = (-1)^s \times 2^{e-1023} \times (1 + f \times 2^{-52}) \quad (0 < e < 2047)$$

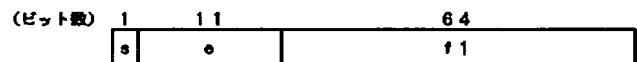
$$= \text{NaN (不定)} \quad (e = 2047, f \neq 0)$$

$$= (-1)^s \times \infty \quad (e = 2047, f = 0)$$

$$= (-1)^s \times 2^{e-1022} (f \times 2^{-52}) \quad (e = 0, f \neq 0)$$

$$= 0 \quad (e = 0, f = 0)$$

(B) 専用プロセッサ2内での76ビット内部フォーマット表現

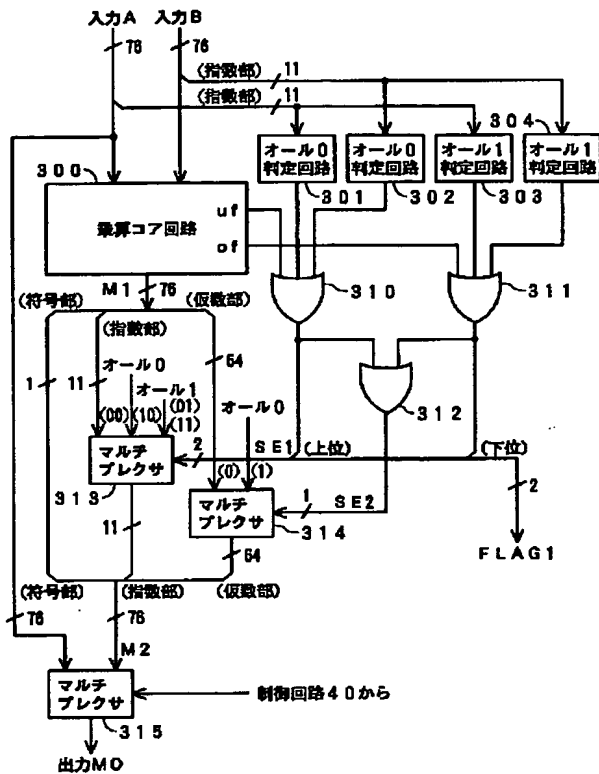


$$v = (-1)^s \times 2^{e-1023} \times (1 + f \times 2^{-64}) \quad (0 < e < 2047)$$

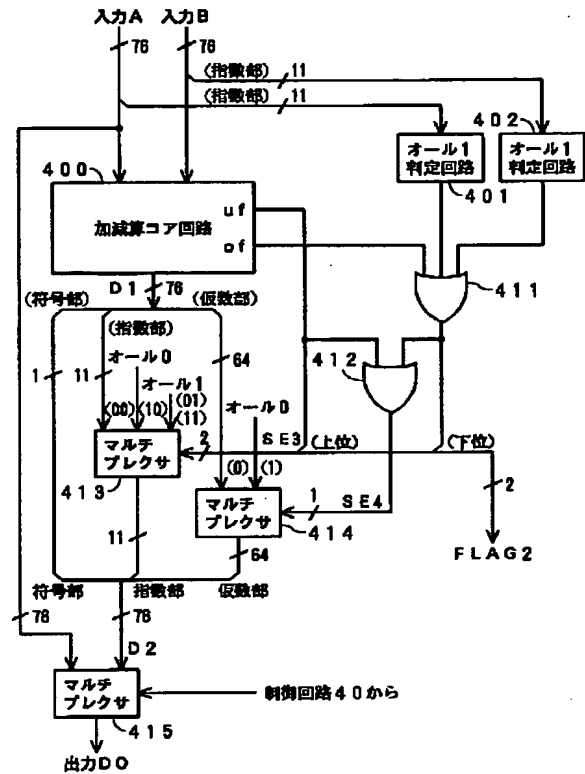
$$= (-1)^s \times \infty \quad (e = 2047)$$

$$= 0 \quad (e = 0)$$

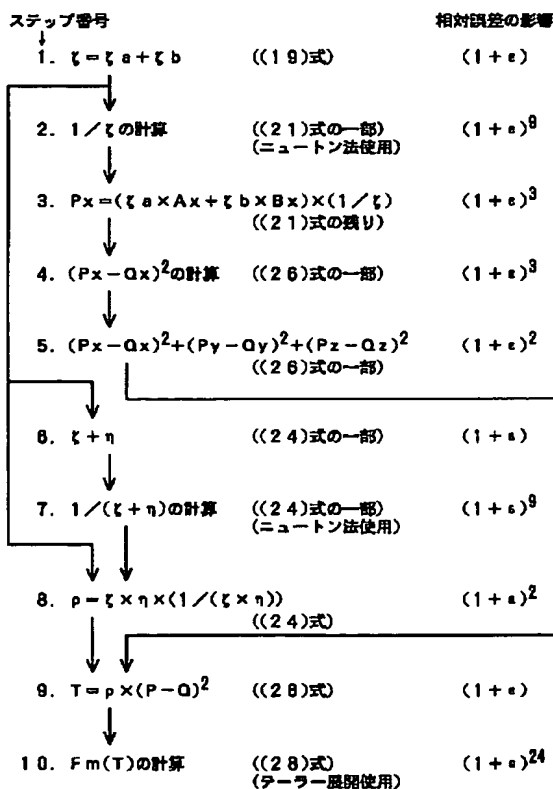
【図 4】



【図 5】



【図 7】



フロントページの続き

(72)発明者 山田 想
神奈川県足柄上郡中井町境430 グリーン
テクなかい富士ゼロックス株式会社内
(72)発明者 宮川 宣明
神奈川県足柄上郡中井町境430 グリーン
テクなかい富士ゼロックス株式会社内
(72)発明者 網崎 孝志
鳥取県米子市旗ヶ崎 8 - 11 - 16

(72)発明者 高島 一
埼玉県大宮市吉野町 1 - 403 大正製薬株
式会社内
(72)発明者 北村 一泰
埼玉県大宮市吉野町 1 - 403 大正製薬株
式会社内
Fターム(参考) 5B013 DD03 DD05
5B022 AA03 BA01 CA01 CA03 CA04
DA04 FA06
5B045 AA07 BB12 GG08 GG11